DOCUMENT RESUME

ED 076 598 TM 002 577

AUTHOR	Wingersky, Marilyn S.; Lord, Frederic M.
TITLE	A Computer Program for Estimating Examinee Ability and Item Characteristic Curve Parameters When There are Omitted Responses.
INSTITUTION	Educational Testing Service, Princeton, N.J.
SPONS AGENCY	National Science Foundation, Washington, D.C.; Office of Naval Research, Washington, D.C. Personnel and Training Research Programs Office.
REPORT NO	ETS-RM-73-2
PUB DATE	Feb 73
NOTE	102p.
EDRS PRICE	MF-$0.65 HC-$6.58
DESCRIPTORS	*Computer Programs; *Information Processing; *Input Output Devices; *Item Analysis; Item Sampling; Scores; Technical Reports; *Test Results

ABSTRACT

The major problem in estimating the item characteristic curve parameters is the large number of parameters. The estimation procedure is considered converged when the parameter estimates become less than the errors of calculation; convergence is a slow process. The computer program is written in Fortran IV, and the only input required are the item responses for each examinee. Final output contains the estimated item parameters and abilities. Examples and formulas for models are included as are item response input and output. (RS)

FILMED FROM BEST AVAILABLE COPY

ED 076598

TM 002 577

RM-73-2

RESEARCH MEMORANDUM

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

A COMPUTER PROGRAM FOR ESTIMATING EXAMINEE ABILITY AND ITEM CHARACTERISTIC CURVE PARAMETERS WHEN THERE ARE OMITTED RESPONSES

Marilyn S. Wingersky
and
Frederic M. Lord

This Memorandum is for interoffice use. It is not to be cited as a published report without the specific permission of the authors.

Educational Testing Service
Princeton, New Jersey
February 1973

Table of Contents

		Page
I.	Introduction	1
	Method	1
	Convergence	2
	Program Considerations	4
	Input	4
	Output	5
	Availability	5
	Disclaimer	6
II.	Formulas and Computational Details	7
	Estimating the Abilities	9
	Scaling	11
	Estimating Item Parameters	11
	Termination of Item Parameter Iterations	18
	Shortening the Stages	19
	Program Termination	19
	References	20
	Appendix A	A1
	Input	A1
	Item Response Input and Output	A3
	Card Input for the Initial Run on a Set of Data	A5
	Card Input for Continuing Runs	A11
	Punched Output	A17
	Printed Output	A19
	Appendix B	B1
	Program Listing	B1

A Computer Program for Estimating Examinee Ability and Item Characteristic Curve Parameters When There Are Omitted Responses[1]

I. Introduction

Given the responses of a group of examinees to a set of items, the computer program, LOGIST, has been developed to estimate the item characteristic curve parameters for each item and the ability of each examinee in terms of Birnbaum's three-parameter logistic model. The parameters are estimated by a method analogous to the maximum likelihood method described by Lord (1968) with the likelihood function modified to handle omits (see Lord, 1973).

The parameters of item characteristic curves are needed for effective design of tailored tests, for comparing the relative efficiency of two tests or of two scoring methods at a specified ability level, and also for computing a scoring weight for each item that will maximize the effectiveness of the test at some specified ability level (see Lord, 1968).

Method

The major problem in estimating the item characteristic curve parameters is the large number of parameters. For each item there are three parameters, the discriminating power, the difficulty, and the lower asymptote of the item characteristic curve, and there is one ability parameter for each examinee. Although the origin and unit of measurement of the abilities are arbitrary, all abilities are estimated. Thus, if we have N

examinees and n items, there are $N + 3n$ parameters to be estimated. In addition, the first derivatives of the log of the criterion function, the likelihood function modified to handle omits, (formulas given in Section II) are nonlinear in the unknown parameters, and a solution must be found using an iterative procedure.

The iterative procedure used is a modified Newton's method. Newton's method applied directly requires solving a system of $N + 3n$ linear equations to get the new parameter estimates. However, it is possible to handle this large number of parameters by the following procedure. At first, the item parameters are held fixed and new examinee abilities estimated, each separately, as the abilities are independent of each other. Using the new ability estimates and holding them fixed, new item parameters are estimated, separately for each item, as the items are independent of each other. This manipulation of the equations first gives us N linear equations, each in one unknown, to solve for new estimates of the ability parameters. Then it gives us n sets of three linear equations in three unknowns to solve for new estimates of the item parameters. The two sections are termed a stage, and the stages are repeated until the procedure converges.

Convergence

The estimation procedure is considered converged when the difference between the parameter estimates becomes less than the errors of calculation, causing the criterion function to fluctuate. This usually takes 30 to 40 stages. This convergence criterion is a strict one--fewer stages can yield quite satisfactory results. It may be essential for convergence to

have a reasonably large number of items, say 40 or more, and a reasonably large number of examinees say 1000 or more.

To achieve convergence it has been necessary to adopt certain rules and invoke certain restrictions. Because of inadequate information, the abilities for examinees with zero scores, for examinees with perfect scores, and for examinees who answered less than one-third of the items are not estimated. Since some of the icc lower asymptotes are poorly determined because of sampling fluctuations, their iterative movements are restricted. In any event, wild fluctuations are avoided by restricting the amount that any parameter may change in a single iteration.

Despite all efforts at control, the estimated discriminating power of one or more items may become unreasonably large. The program may be instructed to remove such items automatically. We suggest that such items be reinserted after over-all convergence and their parameters iterated to convergence while holding all ability estimates fixed.

Even with these rules and restrictions convergence is a slow process. After observing that in many stages some of the parameters change a lot while a majority change only a little, it seemed that the time spent per stage could be reduced by computing new parameter estimates only for those items and abilities for which the previous change in the parameters was greater than two-tenths of the maximum change in the previous stage for that type of parameter. This strategy requires more stages but usually reduces the total computer time for final convergence. Periodically, to update all parameters, then regularly after all changes have become small, all parameters are estimated during a stage.

Program Considerations

The computer program is written in FORTRAN IV with several IBM 360 assembly language routines for bit manipulation, timer access, and non-FORTRAN input and output. On the IBM 360/65, the program compiles in 134K storage and executes in 116K core storage using the overlay feature of 360 OS.

The dimensions of the program allow a maximum of 3000 examinees and 96 items.

Two disk data sets are created from the input data. One contains a record for each examinee of item responses; the other contains a record for each item of the examinee responses. In both sets the responses are in packed binary, 16 responses per word. These data sets are saved for subsequent runs in case convergence is not achieved in the first run. The disk space required for a test of the maximum size allowed by the program is 27 tracks for both data sets.

The following statistics give some idea of the computer time required for running on an IBM 360/65. A test of 90 items and 2926 examinees took approximately 180 seconds per complete stage. A test with 90 items and 994 examinees took approximately 70 seconds per complete stage. The procedure converges in approximately 30 to 40 stages.

Input

The only input required are the item responses for each examinee, coded 01 if correct, 00 if wrong, 10 if omitted, and 11 if not reached. These responses can be in binary form packed 16 responses per word, or in BCD, on tape, disk, or cards. Input options allow for item selection; also examinee selection limited to omitting the first j examinees and

selecting every k -th examinee thereafter. The detailed description is given in the appendix.

Throughout this manual, the term "omitted" does not include items coded "not reached." Generally, all items subsequent to the last indicated response of an examinee are coded "not reached."

Output

The final output contains the estimated item parameters and abilities, printed and punched.

Intermediate printout contains sufficient information for following the progress of the iterations. For each item parameter estimation, the following is printed: its previous value, the current value, the change, whether the parameter was restricted, and also the first derivative of the log of the criterion function. A summary at the end of each stage shows the values of the criterion function, also the maximum change and the maximum derivative for each type of parameter.

At the end of a run all parameters are punched whether or not the procedure converged. If the procedure did not converge, it can be continued where the current run terminated by using the punched output. The appendix explains in detail how to read the printout and continue the process.

Availability

A copy of the program may be obtained upon written request from the authors, Educational Testing Service, Princeton, New Jersey 08540. The user must provide a tape on which the program will be loaded in

80-character card images. The user must specify whether the tape should be blocked, in EBCDIC or BCD, 7-track or 9-track, and the tape density and parity. The tape will be unlabeled.

Disclaimer

Although the program has worked satisfactorily on the data we have tried, no claim is made that the program is free of error.

II. Formulas and Computational Details

The three-parameter logistic model defines the probability that examinee a will get item i correct as

$$(1) \qquad P_{ia} = \text{Prob}(u_{ia} = 1 | a_i, b_i, c_i; \theta_a) = c_i + (1 - c_i)\psi(\ell_{ia})$$

where

$$(2) \qquad \ell_{ia} = Da_i(\theta_a - b_i) \quad ,$$

ψ is the logistic function

$$(3) \qquad \psi(\ell_{ia}) = \frac{1}{1 + \exp(-\ell_{ia})} \quad ,$$

D is a constant approximately equal to 1.7, θ_a is the ability level for examinee a , a_i is the item discriminating power, b_i is the item difficulty, and c_i is the lower asymptote for item i . u_{ia} is 1 if the item was answered correctly and 0 if the item was answered incorrectly.

The program computes estimates of a_i , b_i , c_i , (i = 1,2,...,n), and θ_a , (a = 1,2,...,N), that maximize the criterion function

$$(4) \qquad L^* = \prod_{a=1}^{N} \prod_{i=1}^{n_a} P_{ia}^{v_{ia}} Q_{ia}^{1-v_{ia}}$$

where n is the number of items, N is the number of examinees, n_a is the number of items that examinee a finished. If examinee a responded to item i , $v_{ia} = u_{ia}$. If examinee a omitted item i , $v_{ia} = 1/A$ where A is the number of possible responses for an item. $Q_{ia} = 1 - P_{ia}$. It is shown in Lord (1973) that the estimates obtained from this function

converge for large n to the maximum likelihood estimates that would be found if omits were replaced by random responses.

The parameters that maximize the criterion function minimize the negative of its log, here denoted by F

$$(5) \qquad F = -\log L^* = -\sum_{a=1}^{N} \sum_{i=1}^{n_a} [v_{ia} \log P_{ia} + (1 - v_{ia}) \log Q_{ia}] \quad .$$

Let $\underset{\sim}{z}$ be the vector containing all of the unknown parameters including $\underset{\sim}{\theta}$, the vector of abilities, and $\underset{\sim}{\beta}$, the vector containing the item parameters for all items. To find $\underset{\sim}{z}$ at which F is a minimum, we set the first derivatives of F with respect to $\underset{\sim}{z}$ to zero and solve for $\underset{\sim}{z}$. Since these equations are nonlinear in $\underset{\sim}{z}$, Newton's method, an iterative procedure which requires the matrix of second derivatives of F with respect to $\underset{\sim}{z}$, is used to find the solution. Given an r-th estimate for $\underset{\sim}{z}$, Newton's method computes the $(r + 1)$-th estimate by the following equation

$$(6) \qquad \underset{\sim}{z}^{r+1} = \underset{\sim}{z}^{r} - \left\{ H^{-1} \frac{\partial F}{\partial \underset{\sim}{z}} \right\} \Bigg|_{\underset{\sim}{z}^{r}}$$

where H is the matrix of second derivatives of F with respect to $\underset{\sim}{z}$ and is of order $N + 3n$. To satisfy the necessary condition for a minimum, H must be positive definite at the solution.

By breaking the computation of new estimates into two sections, the problem of inverting an $N + 3n$ matrix is avoided. In the first section, the item parameters are held fixed and H is a diagonal matrix of order

N containing the second derivatives of F with respect to θ. The ability parameters are estimated by solving just one linear equation for each parameter (Section II, p. 9). Each ability may be estimated as many times as desired, or until the difference between two successive estimates is less than a given criterion. The abilities are then scaled (see page 11) to prevent drift and the $\hat{a}$'s and $\hat{b}$'s adjusted accordingly. In the second section the scaled abilities are held fixed and new item parameters estimated, solving a system of three linear equations for each item. The parameters for each item may be estimated as many times as desired or until the difference between two successive estimates of the discriminating power is as small as desired.

The two sections together are termed a stage. The stages are repeated until the changes in the parameters become less than the truncation errors, causing the criterion function to fluctuate.

Estimating the Abilities

The initial estimate for θ_a is the number-right score adjusted for omitted responses and then standardized. The number-right score adjusted for omitted responses is

$$\frac{1}{n_a} \sum_{i=1}^{n_a} v_{ia} \quad .$$

New estimates for the abilities are computed as follows:

From formula 6,

$$\hat{\theta}_a^{r+1} = \hat{\theta}_a^r + \Delta\theta_a \quad ,$$

$$\Delta\theta_a = \left(\frac{\frac{\partial \log L^*}{\partial \theta_a}}{-\frac{\partial^2 \log L^*}{\partial \theta_a^2}} \right) \Bigg|_{\hat{\theta}_a^r, \hat{\underset{\sim}{\beta}}^r} ,$$

$$\frac{\partial \log L^*}{\partial \theta_a} = D \sum_{i=1}^{n_a} \frac{a_i (P_{ia} - c_i)}{(1 - c_i)} \left(\frac{v_{ia} - P_{ia}}{P_{ia}} \right) ,$$

$$\frac{\partial^2 \log L^*}{\partial \theta_a^2} = D^2 \sum_{i=1}^{n_a} \frac{a_i^2 (P_{ia} - c_i)}{(1 - c_i)^2} \frac{Q_{ia}}{P_{ia}} \left\{ \frac{v_{ia} c_i}{P_{ia}} - P_{ia} \right\} .$$

Since Newton's method is unreliable if the second derivative of F is negative, the absolute value of the second derivative is used (see Mathews and Davies, 1971).

Other restrictions imposed to improve convergence are:

(a) If $\hat{\theta}_a^r$ has a different sign from $\hat{\theta}_a^{r-1}$ and $|\Delta\theta_a| > 1$, then set $\hat{\theta}_a^r = \frac{1}{2} \hat{\theta}_a^{r-1} - 1 * \text{sign}(\hat{\theta}_a^{r-1})$.

(b) The amount that $\hat{\theta}_a$ may change is restricted such that $|\Delta\theta_a| \leq |.1 * \hat{\theta}_a^{r-1}| + 2$. If $\Delta\theta_a$ is greater than this amount, then set $\hat{\theta}_a^r = \hat{\theta}_a^{r-1} + \{|.1 * \hat{\theta}_a^{r-1}| + 2\} * \text{sign}(\Delta\theta_a)$.

A message is printed if any adjustments are made in the $\hat{\theta}$'s.

Within a stage for each $\hat{\theta}_a$ new estimates are computed until $\Delta\theta_a$ is less than a user-specified tolerance or the maximum number of iterations has been reached.

Scaling

Since the origin and unit of measurement of the $\hat{\theta}$'s are arbitrary, there are only $N - 2$ independent $\hat{\theta}$'s. However, all N θ's are estimated and then linearly transformed to prevent uncontrolled drift. The $\hat{a}$'s and $\hat{b}$'s are adjusted accordingly. The linear transformation is chosen to make the mean of the $\hat{\theta}$'s zero and the standard deviation one for all $\hat{\theta}$'s inside the range -10 to +10.

Define $\bar{\theta}$ as the mean of the $\hat{\theta}$'s between -10 and +10, and $s_{\hat{\theta}}$ as the standard deviation for the same $\hat{\theta}$'s. Then the standardized $\hat{\theta}$ is

$$\hat{\theta}^*_a = \frac{\hat{\theta}_a - \bar{\theta}}{s_{\hat{\theta}}} \quad .$$

The scaled $\hat{b}_x$'s are

$$\hat{b}^*_i = \frac{\hat{b}_i - \bar{\theta}}{s_{\hat{\theta}}} \quad ,$$

and the scaled $\hat{a}$'s are

$$\hat{a}^*_i = s_{\hat{\theta}}\hat{a}_i \quad .$$

Estimating Item Parameters

Initial estimates. The initial estimates for the item parameters are obtained as follows:

(1) The initial $\hat{a}$'s are set to a constant supplied by the user.

(2) The initial $\hat{b}_i$'s are computed from the conventional item difficulties p_i (proportion of correct answers) by the formula:

$$\hat{b}_i = (\sqrt{1 + \hat{a}_i^2})\, h_i \quad .$$

The h_i are computed from the following formula using the approximation given in Hastings (1955), p. 192.

$$p_i = \frac{1}{2\pi} \int_{h_i}^{\infty} e^{-t^2/2}\, dt \quad ,$$

where

$$p_i = \frac{1}{N_a} \Sigma^*_a v_{ia} \quad .$$

N_a is the number of examinees who reached item i and Σ^*_a indicates the summation over these examinees.

(3) If good estimates of the $\hat{c}$'s are available, they can be used. Otherwise all of the initial $\hat{c}$'s can be set to a chosen constant, or all $\hat{c}$'s may be held at 0, and the parameters for the two parameter logistic model computed.

Different initial estimates may be supplied by the user.

Formulas. From the initial estimates, new parameter estimates are computed for each item as follows:

Let $\hat{a}_i^{r+1} = \hat{a}_i^r + \Delta a_i$, $\hat{b}_i^{r+1} = \hat{b}_i^r + \Delta b_i$, and $\hat{c}_i^{r+1} = \hat{c}_i^r + \Delta c_i$ where the deltas are computed by Newton's formula in the following equation.

$$\begin{vmatrix} \Delta a_i \\ \Delta b_i \\ \Delta c_i \end{vmatrix} = \begin{vmatrix} \frac{\partial^2 F}{\partial a_i^2} & \frac{\partial^2 F}{\partial a_i \partial b_i} & \frac{\partial^2 F}{\partial a_i \partial c_i} \\ \frac{\partial^2 F}{\partial a_i \partial b_i} & \frac{\partial^2 F}{\partial b_i^2} & \frac{\partial^2 F}{\partial b_i \partial c_i} \\ \frac{\partial^2 F}{\partial a_i \partial c_i} & \frac{\partial^2 F}{\partial b_i \partial c_i} & \frac{\partial^2 F}{\partial c_i^2} \end{vmatrix}^{-1} \begin{vmatrix} \frac{\partial \log L^*}{\partial a_i} \\ \frac{\partial \log L^*}{\partial b_i} \\ \frac{\partial \log L^*}{\partial c_i} \end{vmatrix} .$$

The formulas for the first and second derivatives follow, with $\log L = -F$, and Σ^* meaning the summation over the examinees that reached item i.

$$\frac{\partial \log L^*}{\partial a_i} = \frac{D}{(1 - c_i)} \Sigma_a^* \frac{(\theta_a - b_i)(P_{ia} - c_i)}{P_{ia}} (v_{ia} - P_{ia})$$

$$\frac{\partial \log L^*}{\partial b_i} = \frac{-Da_i}{(1 - c_i)} \Sigma_a^* \frac{(P_{ia} - c_i)}{P_{ia}} (v_{ia} - P_{ia})$$

$$\frac{\partial \log L^*}{\partial c_i} = \frac{1}{(1 - c_i)} \Sigma_a^* \frac{v_{ia} - P_{ia}}{P_{ia}}$$

$$\frac{\partial^2 \log L^*}{\partial a_i^2} = \frac{D^2}{(1 - c_i)^2} \Sigma_a^* (\theta_a - b_i)^2 (P_{ia} - c_i) \frac{Q_{ia}}{P_{ia}} \left\{ \frac{v_{ia} c_i}{P_{ia}} - P_{ia} \right\}$$

$$\frac{\partial^2 \log L^*}{\partial b_i^2} = \frac{D^2 a_i^2}{(1 - c_i)^2} \Sigma_a^* (P_{ia} - c_i) \frac{Q_{ia}}{P_{ia}} \left\{ \frac{v_{ia} c_i}{P_{ia}} - P_{ia} \right\}$$

$$\frac{\partial^2 \log L^*}{\partial c_i^2} = \frac{1}{(1 - c_i)^2} \Sigma_a^* \left\{ \left(\frac{v_{ia}}{P_{ia}} - 1 \right) - \frac{v_{ia} Q_{ia}}{P_{ia}^2} \right\}$$

$$\frac{\partial^2 \log L^*}{\partial a_i \partial b_i} = - \frac{D}{(1 - c_i)} \sum_a^* (P_{ia} - c_i)\{(\frac{v_{ia}}{P_{ia}} - 1)$$

$$+ \frac{Da_i(\theta_a - b_i)}{(1 - c_i)} \frac{Q_{ia}}{P_{ia}} (\frac{v_{ia} c_i}{P_{ia}} - P_{ia})\}$$

$$\frac{\partial^2 \log L^*}{\partial a_i \partial c_i} = - \frac{D}{(1 - c_i)^2} \sum_a^* (\theta_a - b_i)(P_{ia} - c_i) \frac{Q_{ia}}{P_{ia}^2} v_{ia}$$

$$\frac{\partial^2 \log L^*}{\partial b_i \partial c_i} = \frac{Da_i}{(1 - c_i)^2} \sum_a^* (P_{ia} - c_i) \frac{Q_{ia}}{P_{ia}^2} v_{ia} \quad .$$

Define S as the vector of first derivatives of $\log L^*$, and T as the matrix of second derivatives of $-\log L^*$. S and T are evaluated at $\hat{a}_i^r$, $\hat{b}_i^r$, $\hat{c}_i^r$, and $\hat{\underset{\sim}{\theta}}^r$ for computing the $(r + 1)$-th parameter estimates for item i.

If T is not positive definite, convergence is questionable. To improve convergence when this condition occurs, a method described by Mathews and Davies (1971) is employed which decomposes the matrix into two triangular matrices LU, where L is lower triangular with 1's on the diagonal and U is upper triangular. Any negative diagonal elements of U are set positive, making the reconstructed matrix positive definite and improving the chances for convergence.

Restrictions on the item parameters. To avoid wild fluctuations in the parameter estimates, the amount that each parameter can change is restricted as follows:

$$|\Delta a_i| \leq |.1 \cdot \hat{a}_i^r| + .2 \quad ,$$

$$|\Delta b_i| \leq |.2 \cdot \hat{b}_i^r| + .4 \quad ,$$

$$|\Delta c_i| \leq \text{CLAMDA} \quad , \text{ where CLAMDA is read in.}$$

Any correction exceeding these limits is set to the limit and the changes for the other parameters adjusted by linear interpolation to be proportionate to the change allowed in the restricted parameter.

A few of the discriminating powers may be poorly determined and may become unreasonably large. The program will automatically remove an item from the estimation procedure if the $\hat{a}_i$ exceeds a maximum of 3 or 4 depending on the user's preference. A minimum value slightly greater than 0 must be specified for $\hat{a}_i$. If an $\hat{a}_i$ attempts to become less than the minimum, it is set to the minimum and the other parameters adjusted.

Since many of the $\hat{c}$'s are poorly determined, any movement is severely restricted. The restrictions are based on an approximation to the standard error of $\hat{c}_i$ denoted by σ_c^*, on an approximate median value for all the $\hat{c}_i$ denoted by $\bar{c}$, on a minimum $c_{min} > 0$, and on a maximum $c_{max} \leq .5$. The restrictions define bounds on $\hat{c}_i$ that $\hat{c}_i$ may exceed only if its standard error is small. The approximation to the standard error is the diagonal element for c_i in the inverse of T. (When the formula for the asymptotic variance has been derived in the same way as the variance for the abilities in Lord (1973) the program will be modified to incorporate it.) An absolute minimum of 0 and maximum of .5 is placed on $\hat{c}_i$. Explicitly the restrictions are as follows, where the r-th estimate of $\hat{c}_i$ is $\hat{c}_i^r = \hat{c}_i^{r-1} + \Delta c_i$, with Δc_i already adjusted so that it does not exceed the maximum allowable change, and where $\hat{c}_i^{r*}$ is the corrected value of $\hat{c}_i^r$.

(1) $\hat{c}_i^{r*} = \bar{c}$ if (1) $|\hat{c}_i^r - \bar{c}| < 1.1\sigma_c^*$ and $\hat{c}_i^{r-1} = \bar{c}$

or (2) $|\hat{c}_i^r - \bar{c}| < .9\sigma_c^*$.

(2) $\hat{c}_i^{r*} = c_{min}$ if $\hat{c}_i^r \leq c_{min}$ and $0 < |c_{min} - \hat{c}_i^r| < 2\sigma_c^*$.

(3) $\hat{c}_i^{r*} = \hat{c}_i^{r-1}$ if $\hat{c}_i^r \geq c_{max}$ and $0 < |\hat{c}_i^r - \hat{c}_i^{r-1}| < 2\sigma_c^*$ and $\Delta c_i > 0$.

(4) $\hat{c}^{r*} = \hat{c}_i^r$ if $\hat{c}_i^r \geq c_{max}$ and $\Delta c_i < 0$.

(5) $\hat{c}_i^{r*} = \hat{c}_i^r$ otherwise.

Having determined $\hat{c}_i^{r*}$, the following actions will be based on its value:

(1) If $|\hat{c}_i^{r-1}| < \hat{c}_i^{r*} < |\hat{c}_i^r$, the corrections to $\hat{a}_i$ and $\hat{b}_i$ are adjusted accordingly by linear interpolation.

(2) If $\hat{c}_i^{r*}$ is not between $\hat{c}_i^{r-1}$ and $\hat{c}_i^r$, the new estimates of $\hat{a}_i$ and $\hat{b}_i$ are accepted as is.

(3) If $\hat{c}_i^{r*} = \hat{c}_i^{r-1}$, new estimates for $\hat{a}_i$ and $\hat{b}_i$ are computed with the c row and column of T and the c element of S deleted.

(4) If $\hat{c}_i^{r*} = \hat{c}_i^{r-2}$, where r - 2 is in this stage, the iterations in this stage are repeated with $\hat{c}_i$ held fixed.

If T is not positive definite, no restriction is placed on $\hat{c}_i$, except that it be bounded by 0 and .5. After T first becomes positive definite, $\hat{c}_i$ is restricted. However, if T again becomes nonpositive definite $\hat{c}_i$ is unrestricted throughout all remaining stages.

Item parameter options and notes. Because nonpositive definite matrices are primarily produced by poorly determined $\hat{c}$'s, the $\hat{c}$'s are usually held fixed at their initial values for the first four or five stages, and

then allowed to vary. However, the user may let them vary from the beginning.

For each item, a parameter code is used to keep track of the number of parameters allowed to vary, also whether $\hat{c}_i$ is restricted or not depending on whether T is positive definite or not. The following table displays values of this code.

Condition of T	$\hat{a}_i$ and $\hat{c}_i$ held fixed	$\hat{c}_i$ held fixed	All three parameters allowed to vary	If $\hat{c}_i$ varies, is its movement restricted or not?
positive definite	1	2	3	restricted
nonpositive definite	-1	-2	-3	not restricted
positive definite after being nonpositive definite	11	12	13	restricted
nonpositive definite after being positive definite after being nonpositive definite. (Once an item has this code, the code is not changed for the rest of the iteration procedure regardless of the condition of T .)	-11	-12	-13	not restricted

The $\bar{c}$, c_{min} , and c_{max} are specified by the user. It may be desirable to change $\bar{c}$ after the $\hat{c}_i$'s have stabilized if the median of the values to which the $\hat{c}_i$'s want to move is not equal to $\bar{c}$. (The median can be determined from the printout.) A good value for c_{max} is 1./A . Possibly c_{min} should be the same distance below $\bar{c}$ as c_{max} is above it.

The minimum value for $\hat{a}_i$ and the maximum value, beyond which the item will be removed, are specified by the user.

Since the initial estimates of the abilities are better approximations than the initial estimates for the item parameters, the first stage should begin by estimating the item parameters. For the remainder of the procedure a stage begins by estimating abilities.

The program contains the option to estimate only the abilities and the item difficulties. Under this option, the scaling consists only of subtracting the mean of the abilities between -10 and 10 from the abilities and from the item difficulties.

Termination of Item Parameter Iterations

The following circumstances determine when the iterations for each item will terminate within a stage:

1. The maximum number of iterations specified by the user has been reached. Because of the restrictions on $\hat{c}_i$, this must be an even number.

2. The relative change in $\hat{a}_i$, $(\Delta a_i/\hat{a}_i)$, is less than some tolerance level. For the first stage this tolerance is specified by the user. In following stages, the tolerance is set to .3 times the maximum relative change over all $\hat{a}_i$ in the previous stage, or .001, whichever is greater.

3. Although the relative change in $\hat{a}_i$ is less than the tolerance, another iteration will be done if the change in $\hat{c}_i$ is greater than or equal to the maximum allowable change or if $\hat{c}_i$ changed and the number of iterations is uneven.

Shortening the Stages

All parameters are estimated in the first stage and every k -th stage, where k is an input parameter, of every run on a set of data. Also, all parameters are estimated whenever the maximum change in the discriminating power in the previous stage is less than some user specified criterion. At other times, only the abilities whose previous change was greater than .2 times (maximum change over all abilities between -2 and +2) are reestimated; item parameters are estimated only for the items that had a parameter in the last stage in which it was estimated change by more than .2 times the maximum change in the previous stage for the same type of parameter.

Program Termination

The program will terminate and punch all parameters for the following reasons.

1. The procedure has converged. This is defined by a decrease in the criterion function after all of the $\hat{c}_i$ have stabilized. At this point all of the first derivatives should be approximately 0, and the thetas should have a mean of 0 and a standard deviation of 1, approximately.

2. The maximum number of stages for this run have been completed.

3. There is not enough time for another complete stage.

4. The maximum number of items have been removed. Items will be removed if $\hat{a}_i$ exceeds the upper limit.

The procedure may be restarted where a previous run ended, using the punched output.

References

Hastings, C., Jr. Approximations for digital computers. Princeton, N.J.: Princeton University Press, 1955. P. 192.

Lees, D. M., Wingersky, M. S., and Lord, F. M. A computer program for estimating item characteristic curve parameters using Birnbaum's three-parameter logistic model. ONR Technical Report, Contract No. N00014-69-C-0017. Princeton, N.J.: Educational Testing Service, 1972.

Lord, F. M. An analysis of the Verbal Scholastic Aptitute Test using Birnbaum's three-parameter logistic model. Educational and Psychological Measurement, 1968, 28, 989-1020.

Lord, F. M. Estimation of latent ability and item parameters when there are omitted responses. Research Bulletin 73-00. Princeton, N.J.: Educational Testing Service, 1973, in press.

Mathews, A. and Davies, D. A comparison of modified Newton methods for unconstrained optimisation. The Computer Journal, 1971, 14, 293-294.

Footnote

[1]This program and write-up revise, extend, and replace the earlier version and its write-up (Lees, Wingersky, and Lord, 1972). The authors are indebted to Diana Lees for the earlier program which made this version possible. Research reported in this paper has been supported by grant GB-32781X from the National Science Foundation and by contract no. N00014-69-C-0017 with the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research.

Appendix A

Input

There are two input data sets; one is on cards and specifies the options and necessary constants, the other is on unit 10 and contains the item responses.

The program allows the user the following options:

1. The item responses may be in binary, or BCD on disk, tape, or cards.

2. Examinees may be selected by omitting the first j examinees and accepting every k-th one thereafter, where j and k are chosen by the user.

3. Up to 100 specific examinees may be omitted.

4. The initial starting values for any of the types of parameters may be read rather than computed.

5. The amount of printed output is variable.

6. All of the $\hat{c}$'s may be held fixed for the whole procedure or for only the first k (specified by the user) stages.

7. Only the abilities and the item difficulties may be estimated.

The user must specify the following:

1. The items to be selected from the item response input.

2. For estimating abilities, the maximum number of iterations per ability per stage as well as the criterion for $\Delta\theta$ for determining whether another iteration will be done.

3. For estimating the item parameters, the maximum number of iterations per item per stage and the criterion for the relative change in $\hat{a}_i$ used in the first stage to determine whether another iteration will be done.

4. The starting value for all of the $\hat{a}$'s.

5. The maximum allowable change for $\hat{c}_i$ in an iteration.

6. The "median value" $\bar{c}$.

7. The probability of getting an item right by guessing at random.

8. The bounds on the range of $\hat{c}_i$, such that these limits will only be exceeded if the approximate standard error is small.

9. The criterion for removing an item. If $\hat{a}_i$ exceeds this value the item will be removed.

10. The bounds on the range of $\hat{a}_i$. The lower limit should be close to 0, the upper limit no less than the criterion for removing an item.

11. A criterion value that determines whether or not all of the parameters will be estimated in the next stage. If the maximum relative change for $\hat{a}_i$ is less than this value, all of the parameters will be estimated in the next stage.

12. The maximum amount of time for the run in seconds.

When resuming computations using the punched output of a previous run, one may modify the output by changing abilities, adding or deleting items, changing item parameters or changing the number of parameters allowed to vary.

Item Response Input and Output

The item responses may be on disk, tape, or cards in BCD or binary, one record per examinee. The item responses are on unit 10. Examinees are numbered in the order that they are read, with those examinees skipped at the beginning of the data set or those skipped when selecting every k -th person not included in the numbering. Of those examinees numbered, up to 100 may be excluded. Examinees who have 0 scores, perfect scores or who answer less than a third of the items are not included in the estimation procedure.

The item responses are coded in a two-digit code: 01 if the item is correct, 00 if wrong, 10 if omitted, 11 if not reached. In binary, these responses are packed 16 responses per word. The cards in the setup specify which items are to be selected from the input record, considering the items numbered consecutively from 1 to the total number of items in the record.

The first run on a set of data reads the item response input and creates two work data sets to be used in this run and continuing runs on the same data set. The original item response data set is usually not used in subsequent runs.

The first work data set, on unit 11, contains the selected item responses packed 16 per word for each examinee and is used in the ability estimation section. The record length equals the block length which is 2400 bytes or 600 IBM 360 machine words. The number of blocks required for a given set of data is computed as follows, where n is the number of items and N is the number of people selected:

The number of words per person is $(n - 1)/16 + 1$.

The number of people per block is 600 divided by the number of words per person.

The number of blocks is N divided by the number of people per block plus one.

The data set on unit 12 contains the item responses of all examinees for each selected item packed 16 per word ordered by item and is used in the item parameter estimation section. The record length equals the block length which is 3008 bytes or 752 IBM 360 machine words. The number of blocks required for a given data set is computed as follows:

The number of words per item is $(N - 1)/16 + 1$.

The number of items per block is 752 divided by the number of words per item.

The number of blocks is n divided by the number of items per block plus one.

Card Input for the Initial Run on a Set of Data

Card 1 — Title card

Cols. — FORMAT(15A4,2X,I3,I7)

1-60 — Title for the data

63-65 — n = number of items, $n \leq 96$

66-72 — N = number of examinees, $N \leq 3000$

Card 2 — Options card

Cols. — FORMAT(16I5)

1-5 — MAXST = Maximum number of stages for this run.

6-10 — MAXIT = Maximum number of iterations per item per stage. This must be an even number.

11-15 — = 1 For each iteration of each $\hat{\theta}_a$, print the first and second derivatives, the change in $\hat{\theta}$ and the new $\hat{\theta}_a$.

= blank Otherwise.

16-20 — = 0 Initial $\hat{c}_i = 0$, $i = 1,2,\ldots,n$ and $\hat{c}_i$ is held fixed for all items.

= 1-9 Number of choices per item. The initial $\hat{c}_i$ = 1/(the number of choices per item), $i = 1,2,\ldots,n$.

= 100 Set the initial $\hat{c}$'s to $\bar{c}$.

= 1000 Read in the initial $\hat{c}$'s. See card set 9.

20-25 — NIT = 0 Begin this run with the estimation of the item parameters, usually done for the first run on a set of data.

Card 2 (continued)

20-25 = 1 Begin this run with the estimation of abilities, usually done when continuing iterations on a set of data.

26-30 NØT = Number of examinees to be skipped at the beginning of the data set.

31-35 INPUT = 0 Item responses are in binary on unit 10.

= 1 Item responses are in BCD on unit 10.

36-40 NSKIP = Number of examinees to be skipped when reading the item response data set, after eliminating the first "NØT" examinees. If NSKIP > 0, then take every "NSKIP"-th examinee, e.g., if NSKIP = 4 then 4th, 8th, 12th, etc. examinees will be used. If NSKIP = 1 and NØT = 0, the first N examinees will be used.

41-45 NC < 0 Hold all $\hat{c}_i$ fixed until the |NC|-th stage. Then allow them to vary.

= 0 Vary all item parameters unless NCH = 0 , then hold all $\hat{c}$'s fixed.

= 1 Hold all $\hat{c}$'s fixed.

= 2 Hold all $\hat{a}$'s and $\hat{c}$'s fixed. Compute only the abilities and item difficulties.

46-50 IIAST = Number of last stage in which the vector S and matrix T will be printed.

Card 2 (continued)

51-55 IT = 0 Compute initial estimates for $\hat{\theta}_a$, a = 1,2,...,N.

= 1 Read in initial estimates for $\hat{\theta}_a$, a = 1,2,...,N. See card set 6.

56-60 IA = 0 Set $\hat{a}_i$ = AINIT (value punched on the following card) for all items.

≠ 0 Read in the initial estimates for $\hat{a}$'s. See card set 7.

61-65 IB = 0 Compute initial estimates for $\hat{b}_i$, i = 1,2,...,n.

≠ 0 Read in the initial estimates for $\hat{b}_i$, i = 1,2,...,n. See card set 9.

66-70 IY = 0 Estimate all parameters in every stage.

≠ 0 Estimate all parameters in every IY-th stage.

71-75 ITONE = 0 Do only one iteration per stage for each $\hat{\theta}_a$.

> 0 Number of iterations per stage for each $\hat{\theta}_a$.

Card 3 Specification Card

Cols. FORMAT(F6.0,12F6.2)

1-6 TIMAX - The maximum time in seconds for this run.

7-12 TOL1 - If $\Delta\theta_a$ is greater than TOL1, another iteration will be done unless the maximum number of iterations has been reached.

13-18 CLAMDA - The maximum allowable change in $\hat{c}$.

19-24 CBAR - The median $\hat{c}$.

25-30 CLO - The minimum value that $\hat{c}_i$ may not go below unless it has a small standard error.

Card 3 (continued)

31-36 CHI - The maximum value that $\hat{c}_i$ may not exceed unless it has a small standard error.

37-42 TOL - If the relative change in a_i is greater than TOL, another item parameter iteration will be done unless the maximum number of iterations has been reached.

43-48 CGUESS - The probability of getting an item correct by guessing at random. This is $1/A$.

49-54 AINIT - The initial value for $\hat{a}_i$, $i = 1,2,\ldots,n$.

55-60 AIØ - The minimum value for $\hat{a}_i$ should be around .01.

61-66 AHI - The maximum value for $\hat{a}_i$ should be greater than ALIM.

67-72 ACRIT - The criterion value of the relative change in $\hat{a}_i$ such that if the maximum relative change in $\hat{a}_i$ is less than this amount, all parameters will be estimated in the next stage.

73-78 ALIM - The criterion value for $\hat{a}_i$ for removing an item. If $\hat{a}_i$ exceeds this value, item i will be removed.

Card 4

Cols. FØRMAT(16I5)

1-5 MAXØUT - The maximum number of items which may be removed if $\hat{a}_i > \text{ALIM}$.

6-10 ICKA > 0 Do not remove items on the basis of $\hat{a}_i$ exceeding ALIM.

= 0 Remove item i if $\hat{a}_i > \text{ALIM}$.

Card 4 (continued)

11-15 NKILL = Number of examinees to be excluded from the estimation procedure.

16-20 KILL(1) - Number (see paragraph 1, p. A3) of the first examinee to be excluded.

21-25 KILL(2) - Number of second examinee to be excluded.

⋮ ⋮

KILL(NKILL) - Number of NKILL-th examinee to be excluded.

Card 5 Item Selection Card

Cols. FORMAT(26I3)

1-3 ITEMNO(1) - Number of first item to be selected.

4-6 ITEMNO(2) - Number of second item to be selected.

⋮

ITEMNO(n) - Number of n -th item to be selected. This card controls the selection of items on the input record and assumes the items are numbered consecutively from 1 to the total number of items in the record.

Card(s) 6 Omit if IT (col. 55 of card 2) does not equal 1

card a Set column 2 to 0. The rest of the card is blank.

card b FORMAT for reading $\hat{\theta}$'s, which must be read as floating point numbers.

card(s) c $\hat{\theta}_a$ for a = 1,...,N punched in format given in card b.

Card(s) 7 Initial values for $\hat{a}_i$ (Omit if IA = 0 on card 2)

card a cols. 1-80 FORMAT for reading the $\hat{a}$'s which must be read as floating-point numbers.

Card 7 (continued)

card(s) b — $\hat{a}(i)$, (i = 1, ...,n) - Punched in the format specified in card a.

Card(s) 8 — Initial values for $\hat{b}_i$ (Omit if IB = 0 on card 2)

card a — cols. 1-80 — Format for reading the $\hat{b}$'s which must be read as floating-point numbers.

card(s) b — $\hat{b}(i)$, (i = 1,...,n) - Punched in the format specified in card a

Card(s) 9 — Initial values for $\hat{c}_i$ (Omit if col. 16-20 on card 2 $\neq$ 1000)

$\hat{c}(i)$, (i = 1,...,n) - Punched in 8F10.7 format.

Card Input for Continuing Runs

The setup for continuing the estimation procedure consists of the first four cards for the initial run with the following changes:

Card 2

col. 25 = 1 Begin the run with the estimation of abilities.

55 = 2 Read in the output of a previous run. The work data sets have already been created.

= 3 Read in the output of a previous run and recreate the work data sets (usually not necessary).

Any of the other parameters may be changed.

If no changes are to be made to the output being read, the following cards are placed after the first four cards.

Card 5

col. 2 = 1

Card(s) set 6 The punched output from the previous run.

All following runs only require replacing the punched output.

However, if it is necessary to modify the punched output, the following cards are placed after card 4 of the initial run setup. (The following is long and detailed. If it does not apply, skip to the description of punched output.)

The modification card, card 5 of the continue setup contains an indicator for each change that will be 1 if the change is required, blank if no change is required. A description of the card follows.

Card 5

Cols.	FØRMAT(40I2)
2	INP = 1 $\hat{\theta}$'s are punched in hexadecimal in A4 format.
	= 0 $\hat{\theta}$'s punched in E format.
4	JCHTH = 1 Change some $\hat{\theta}$'s read.
6	MI(2) = 1 Add more items than read.
8	MI(3) = 1 Remove some of the items from the input.
10	MI(4) = 1 Change the $\hat{a}$'s, $\hat{b}$'s and $\hat{c}$'s for some items.
12	MI(5) = 1 Change the $\hat{a}$'s for some of the items.
14	MI(6) = 1 Change the $\hat{b}$'s for some of the items.
16	MI(7) = 1 Change the $\hat{c}$'s for some of the items.
18	MI(8) = 1 Change the parameter code for some of the items.
20	MI(9) = 1 Change the parameter code for all items.

Card Set 6 The punched output from the previous run.

Card Set 7 Change $\hat{\theta}$'s. (Omit if col. 4 of card 5 = 0)

	Cols.	
card a	1-5	NØUT - Number of $\hat{\theta}$'s to be changed
card b		4(I5,E15.8)
	1-5	Number* of first $\hat{\theta}_a$ to be changed.
	6-20	New value for that $\hat{\theta}_a$.
	21-25	Number* of 2nd $\hat{\theta}_a$ to be changed.
	26-40	New value for that $\hat{\theta}_a$.

*See paragraph 1, p. A3.

Card Set 8 Add items (Omit if col. 6 of card 5 = 0)

	Cols.	
card a	1-5	NOUT - number of items to be added.
card(s) b		(I5,3E15.8,I5) A card for each item in order by item number.
	1-5	Number* of item to be added.
	6-20	Initial $\hat{a}$ value for that item.
	21-35	Initial $\hat{b}$ value for that item.
	36-50	Initial $\hat{c}$ value for that item.
	51-55	ITEST; parameter code.**

Card Set 9 Remove items (Omit if col. 8 of card 5 = 0)

Cols.	(16I5)
1-5	NOUT - Number of items to be removed.
6-10	Number* of 1st item to be removed.
11-15	Number* of 2nd item to be removed.
etc.	etc.

Card Set 10 Change all of the parameters for some items (omit if col. 10 of card 5 = 0)

	Cols.	
card a	1-5	NOUT - Number of items changed.

---

*The number of the item is its item number in the item response data set on unit 10.

**Parameter code = 1 if only b_i is allowed to vary.
= 2 if a_i and b_i vary, but c_i is held fixed.
= 3 if all parameters are allowed to vary.

Card Set 10 (continued)

card(s) b	(I5,3E15.8.I5)	A card for each item ordered by item number
	1-5	Number* of item for which the item parameters are to be changed.
	6-20	New $\hat{a}$.
	21-35	New $\hat{b}$.
	36-50	New $\hat{c}$.
	51-55	ITEST; parameter code.**

Card Set 11 Change some of the $\hat{a}$'s (Omit if col. 12 of card 5 is 0)

	Cols.	
card a	1-5	NOUT - Number of items for which the $\hat{a}$'s are to be changed.
card(s) b		(4(I5,E15.8))
	1-5	Number* of item for which $\hat{a}$ is to be changed.
	6-20	New value for $\hat{a}$.
	21-25	Number* of next item for which $\hat{a}$ is to be changed.
	26-40	New value for that $\hat{a}$.
	etc.	etc.

---

*The number of the item is its item number in the item response data set on unit 10.

**See ** footnote, p. A13.

Card Set 12 Change some of the $\hat{b}$'s (Omit if col. 14 of card 5 is 0)

	Cols.	
card a	1-5	NOUT - Number of $\hat{b}$'s to be changed. (4(I5,E15.8))
card(s) b	1-5	Number* of 1st item for which $\hat{b}$ is to be changed.
	6-20	New value of $\hat{b}$ for that item.
	21-25	Number* of 2nd item for which $\hat{b}$ is to be changed
	26-40	New value of $\hat{b}$ for that item.
	etc.	etc.

Card Set 13 Change some of the $\hat{c}$'s (Omit if col. 16 of card 5 is 0)

	Cols.	
card a	1-5	NOUT - Number of $\hat{c}$'s to be changed. (4(I5,E15.8))
card(s) b	1-5	Number* of 1st item for which $\hat{c}$ is to be changed.
	6-20	New value of $\hat{c}$ for that item.
	21-25	Number* of 2nd item for which $\hat{c}$ is to be changed.
	26-40	New value of $\hat{c}$ for that item.

---

*The number of the item is its item number in the item response data set on unit 10.

Card Set 14 Change the parameter codes** for some of the items (Omit if col.18 of card 5 is 0)

	Cols.	
card a	1-5	NOUT - Number of parameter codes to be changed.
card b		(16I5)
	6-10	Item number* of first parameter code to be changed.
	11-15	New parameter code** for that item.
	16-20	Item number* of second parameter code to be changed.
	21-25	New parameter code** for that item.
	etc.	etc.

Card Set 15 Change the parameter codes** for all of the items (Omit if col. 20 of card 5 is 0)

Cols.	
1-5	New parameter code** for all items.

---

*The number of the item is its item number in the item response data set on unit 10.

**See ** footnote, p. A13.

Punched Output

Card 1	FORMAT(15A4,2I3,2I6)
1-60	Title
61-63	Number of items read
64-66	Number of items for which parameters were estimated
67-72	Number of examinees read
73-78	Number of abilities estimated
Card 2	FORMAT(5A4,I5)
1-20	Date and time of run
21-25	The number of the last stage in this run
Card 3	FORMAT(A8,4A4)
1-8	Product of final ability estimates (floating point)
9-12	Exponent of the product (integer)
13-16	Mantissa of the criterion function for the last stage in the run (floating point)
17-20	Exponent of the criterion function for the last stage (integer)
21-24	TOLA - The tolerance for the relative change in a_i for determining whether another iteration should be done.
Card(s) 4	FORMAT(I4,19A4)

This card set contains the final abilities, θ_a, $a = 1,2,\ldots,N$ for all examinees. If an examinee's ability was not estimated, it is set to -999999. The first four columns contain the sequence number of the card starting with 0001; the remaining 19 fields contain abilities.

Card(s) 5 FORMAT(I4,18A4)

The item parameters $\hat{a}_i$, $\hat{b}_i$, and $\hat{c}_i$ for all items are punched six items per card in cols. 5-76 for as many cards as necessary. The first four columns contain the card sequence number starting with 1001. The parameters for items that have been removed from the estimation procedure are included in the punched output.

Card(s) 6 FORMAT(16I5)

The parameter code for each item. (See p. 16 for a description of the codes.

Card(s) 7 FORMAT(26I3)

The item numbers* for the items in the output.

Card(s) 8 FORMAT(16I5)

The first field contains the number of examinees omitted from the estimation procedure, followed by the sequence numbers** (in numerical order) of the examinees omitted.

Card(s) 9 FORMAT(16I5)

The first field contains the number of items that have been removed followed by the item numbers* (in numerical order) of the removed items.

---

*The number of the item is its item number in the item response data set on unit 10.

**See paragraph 1, p. A3.

Printed Output

Printout at the Beginning

1. The options selected and constants specified.

2. If a continuing run, the title, number of items read, number of items estimated, number of examinees read, number of examinees whose abilities were estimated, and the time and date of previous run.

3. The sequence numbers of examinees removed because they had a 0 or perfect score, on request, or because they failed to answer over a third of the items.

4. If the input is binary and this is the first run on a set of data, the first 10 input records are printed in hexadecimal.

5. If this is the first run, and the initial ability estimates are computed, the number-right scores adjusted for omitted items, their mean and variance.

6. Title, number of items read, number of items estimated, number of examinees read, number of abilities estimated, time and date for this run.

7. The proportion of correct responses adjusted for omits for each item, if this is the first run and initial estimates for the item difficulties are being computed.

8. The beginning estimates for $\hat{a}_i$, $\hat{b}_i$, and $\hat{c}_i$ and the parameter code for each item. If this is the first run, the number of people who reached each item and the number of people who omitted each item are also printed. (Only the people whose abilities are being estimated are included.)

9. The beginning estimates for the abilities.

Printout from the Abilities Estimation Section

1. If the detailed printout for the abilities is requested, $\frac{\partial \log L^*}{\partial \theta_a}$ and $-\frac{\partial^2 \log L^*}{\partial \theta_a^2}$, the change in $\hat{\theta}_a$, and the new estimate for $\hat{\theta}_a$ for each examinee. If the $-\frac{\partial^2 \log L^*}{\partial \theta_a^2}$ was negative, four asterisks are printed at the end of the line.

2. If condition (a) described in the restrictions on $\hat{\theta}_a$ occurs, the message "Theta will be set to," the value $\hat{\theta}_a$ is set to, and the amount that $\hat{\theta}_a$ wanted to change.

3. If condition (b) described in the restrictions on $\hat{\theta}_a$ occurs, a line that $\hat{\theta}_a$ has been adjusted, followed by a line containing the examinee number, the iteration number, the change in $\hat{\theta}_a$, the new estimate for $\hat{\theta}_a$, $\frac{\partial \log L^*}{\partial \theta_a}$ and $-\frac{\partial \log L^*}{\partial \theta_a^2}$, and, if $-\frac{\partial^2 \log L^*}{\partial \theta_a^2}$ was negative, four asterisks.

4. If not all of the abilities are being estimated, the cutoff point such that, if the previous change in $\hat{\theta}_a$ was less than this value, the ability will not be estimated.

5. The number of abilities estimated in this stage.

6. The number of examinees for which $-\frac{\partial^2 \log L^*}{\partial \theta_a^2}$ was negative.

7. The maximum change for $\hat{\theta}$'s between -2 and +2.

8. The criterion function computed from the values of the parameters at the beginning of this stage.

9. A message that the criterion function has increased, if it has.

Scaling Printout

1. The mean and standard deviation for the $\hat{\theta}$'s between -10 and +10, as well as the number of $\hat{\theta}$'s in this range.

2. The examinee number and corresponding $\hat{\theta}_a$ for all $\hat{\theta}$'s outside the range of -3 to +3.

3. The scaled $\hat{a}$'s and $\hat{b}$'s.

4. The time for estimating abilities.

Printout for the Item Parameter Estimations

1. The criterion value for the relative change in a_i for doing another item iteration.

2. If not all parameters are being estimated, the cutoff values for the relative Δa_i , Δb_i , and Δc_i , such that if none of the previous changes in the parameters is greater than its respective cutoff value, the estimation of the item parameters will be skipped in this stage.

3. For each item parameter iteration, the first iteration contains from two to four lines. All lines contain the item number. All but the first contain the iteration number. The first line is omitted in subsequent iterations.

   a. The first line contains $\hat{a}_i^{r-1}$, $\hat{b}_i^{r-1}$, $\hat{c}_i^{r-1}$, where this is the r -th iteration.

   b. The last line contains

      i. A code indicating how many parameters were estimated.

         1 only $\hat{b}_i$

         2 $\hat{a}_i$ and $\hat{b}_i$

         3 $\hat{a}_i$, $\hat{b}_i$, and $\hat{c}_i$

ii. $\frac{\partial \log L^*}{\partial a_i^{r-1}}$, $\frac{\partial \log L^*}{\partial b_i^{r-1}}$, $\frac{\partial \log L^*}{\partial c_i^{r-1}}$, $\hat{a}_i^r$, Δa_i , $\hat{b}_i^r$, Δb_i , $\hat{c}_i^r$, Δc_i , σ_c^* .

iii. If T was not positive definite, the code is negative, σ_c^* is enclosed in parentheses, and four asterisks are printed at the end of the line.

c. If $\hat{c}_i$ had to be held fixed at its previous value due to any of its many restrictions, the second line contains the values the parameters wanted to be: $\hat{a}_i^r$, Δa_i , $\hat{b}_i^r$, Δb_i , $\hat{c}_i^r$, '*' , Δc_i , σ_c^* , 'c' , and $\hat{c}_i^{r*}$ (the value $\hat{c}_i$ is fixed at). $\hat{c}_i^r$ is set to $\hat{c}_i^*$, and new estimates computed for $\hat{a}_i$ and $\hat{b}_i$. An '*' is printed beside Δc_i if it is greater than the maximum allowable change.

d. If any adjustments were necessary other than holding $\hat{c}_i$ fixed, a line containing the unadjusted values of $\hat{a}_i^r$, Δa_i , $\hat{b}_i^r$, Δb_i , $\hat{c}_i^r$, Δc_i , with an '*' printed to the right of the value or values that necessitated the adjustment. For example, an '*' to the right of Δb_i means that Δb_i is greater than the maximum allowable change in $\hat{b}_i$.

e. If, in two iterations, $\hat{c}_i$ changes and then moves back to its original value, the iterations are restarted holding $\hat{c}_i$ fixed. When this occurs, line 1 is repeated with an '*' in the leftmost column.

f. If the user so requests, the matrix T and vector S are printed for each iteration.

4. At the end of the item estimation section are printed the new item parameters, the amount that each parameter differs from its value at the end of the last stage, the parameter codes, the number of examinees who reached each item, and the number of examinees who omitted each item (only those examinees whose abilities are being estimated are included).

Printout at the End of A Stage

1. The maximum first derivative of the criterion function for each type of item parameter and the corresponding item number. If any $\hat{c}_i$'s were held fixed, their derivatives are not included in determining the maximum.

2. The maximum first derivative of the criterion function with respect to $\hat{\theta}_a$ and the corresponding examinee number.

3. The maximum change in an iteration for $\hat{a}_i^*$, $\hat{b}_i$, $\hat{c}_i$, with the corresponding item numbers, and for $\hat{\theta}_a$ with the corresponding examinee number.

4. The maximum total change in the stage for $\hat{a}_i^*$, $\hat{b}_i$, $\hat{c}_i$, with the corresponding item numbers, and for $\hat{\theta}_a$ with the corresponding examinee number.

5. The maximum number of iterations in estimating items and in estimating abilities.

6. The criterion function computed from the parameters at the beginning of the item estimation section.

7. Time for the item parameter estimation section.

8. Time for the stage, and stage number.

---

*This is the relative change for the a 's.

Printout at the End of Run

1. Final estimated abilities
2. Summary containing for each stage
   a. Stage number
   b. Criterion function computed after item estimation in the previous stage and after ability estimation in this stage.
   c. The information listed as 1 through 4 above.
3. The total time for the run.

Appendix B

Program Listing*

*This program may undergo future revisions, and we recommend that anyone wanting to use it obtain the updated program from the authors.

```
C                              LOGIST
C      THIS PROGRAM ESTIMATES THE LOGISTIC ITEM CHARACTERISTIC CURVE
C         PARAMETERS USING THE LIKELIHOOD FUNCTION MODIFIED TO HANDLE
C         OMITS
C      "NOT REACHED" ITEMS ARE NOT USED IN THE ESTIMATION PROCEDURE
C                                                                      LOGI   3
       COMMON AHI,ALIM,ALO,ACRIT  ,CBAR,CGUESS,CHI,CHOICE,CLAMDA,CLO,
      1ICKA,ILAST,INDIC,ISTAGE,          JSTAGE,LITTLN,MAXOUT,MAXST,
      5MAXIT,IWILL,ITONE,NOUT,NRECPI,OLDSUM,KEXP,
      2N,NC,NIPREC,NIT,NKILL,NLB,NLP,NPPREC,    TIMAX,    TOL1,TOLA,
      3TOTAL,A(100),B(100),C(100),KILLS(100),KDO(100),ITEMNO(100),
      4IY,NKITE,KITE(100),TITLE(15),DATER(5),THETA(3000)
       REAL*8 TITLE
       COMMON/FSTAT/SPACE(32)
       DO 3 J=1,32
       SPACE(J)=0.
     3 CONTINUE
       CALL SUB1
       CALL SUB2
     1 CALL EXIT
       END
       SUBROUTINE SUB1
C      ITERATES ON PARAMETERS SELECTED ON SIZE OF LAST CHANGE
       COMMON AHI,ALIM,ALO,ACRIT  ,CBAR,CGUESS,CHI,CHOICE,CLAMDA,CLO,
      1ICKA,ILAST,INDIC,ISTAGE,          JSTAGE,LITTLN,MAXOUT,MAXST,
      5MAXIT,IWILL,ITONE,NOUT,NRECPI,OLDSUM,KEXP,
      2N,NC,NIPREC,NIT,NKILL,NLB,NLP,NPPREC,    TIMAX,    TOL1,TOLA,
      3TOTAL,A(100),B(100),C(100),KILLS(100),KDO(100),ITEMNO(100),
      4IY,NKITE,KITE(100),TITLE(15),DATER(5),THETA(3000)
       DIMENSION Z(600),ZZ(3760),              FMT(10),THETT(19)
      2,MI(9),QC(5),ITEST(100),NB(100),THTA(25),WHY(5),  NOM(100),
      1KILL(100),THMED(100),QCFR(100),NPPI(100)
       REAL*8 INPUTB(2)
       DATA INPUTB/'BINARY','BCD'/
       EQUIVALENCE(NB,B)
       REAL*8 RPROD,PROD,FMT
       KIND=0
       IB=1
       CALL SETCLK
       WRITE(6,2899)
 2899 FORMAT('1  LOGISTIC PROGRAM          ITEMS NOT REACHED ARE NOT INC
     1LUDED IN THE COMPUTATIONS')
       CALL DATE(DATER)
     1 CALL RESCLK (1,TIME)                                            LOGI  29
       READ (5,1000) TITLE,LITTLN,N                                    LOGI  31
 1000 FORMAT (15A4,2X,I3,I7)
 2000 FORMAT('1',15A4,2X,'TOTAL NO. OF ITEMS = ',I3,' NO OF ITEMS ESTIMA
     1TED = ',I3/110X,5A4/          10X,                ' NO. OF EXAMI
     1NEES READ IN = ',I8,'  NO. OF THETAS TO BE ESTIMATED ',I8)
       READ (5,1001) MAXST,MAXIT,IWILL,NCH,NIT,NOT,INPUT,NSKIP,NC,ILAST, LOGI  36
```

```
     1IT,IA,IB,IY,ITONE
 1001 FORMAT (16I5)                                                    LOGI  38
      CHOICE=NCH                                                       LOGI  39
C                                                                      LOGI  40
COL  5MAXST = MAXIMUM STAGES                                           LOGI  41
COL 10 MAXIT = MAXIMUM ITERATIONS FOR ITEM PARAMETERS
COL 15 IWILL  =1 PRINT THETA INFORMATION FOR EACH ITERATION
COL 20 NCH=0 ALL C = 0 AND ALL C ARE HELD FIXED
C         =1-10        ALL C SET TO 1/NCH
C         =100 SET INITIAL C TO CBAR
C         =1000 READ IN INITIAL C
COL 25NIT    = 0 SKIP FIRST HALF OF STAGE 1, ITERATE ON ITEM PARAMETERS LOGI  47
C                  UNTIL EITHER A GREATER THAN 1 OR ABS(DEL A) LESS THAN LOGI  48
C                  0.01 - THEN DO ONLY 1 ITERATION ON THETA THEREAFTER   LOGI  49
COL 30NOT = NUMBER OF EXAMINEES TO BE SKIPPED AT THE BEGINNING OF TAPE LOGI  50
C         INPUT IS ON DATASET IN BINARY FORMAT FOLLOWED BY
C              LAST ITEM REACHED-- DATASET SORTED BY LAST ITEM REACHED
COL 35    =0 INPUT IS BINARY
C         =1 INPUT IS BCD
COL 40NSKIP = NO. OF EXAMINEES TO BE SKIPPED WHEN READING TAPE         LOGI  55
C               IF +  SKIP EVERY (NSKIP-1) PEOPLE                      LOGI  56
C          AFTER SKIPPING THE NUMBER SPECIFIED BY NOT - SEE COL 30      LOGI  57
COL 45  NC=0  VARY C
C         =1 HOLD ALL C CONSTANT
COL 50ILAST = LAST STAGE IN WHICH EACH ITERATION WILL BE PRINTED       LOGI  61
COL 55IT     = 0 USE NO. RIGHT NORMALIZED AS INITIAL THETAS            LOGI  62
C      = 1 READ IN INITIAL ESTIMATES OF THETAS IN E FORMAT
C       = 2 READ IN INITIAL PARAMETERS AS OUTPUT FROM PREVIOUS RUN
COL 60IA     - SAME AS ABOVE FOR A'S                                   LOGI  65
COL 65IB     - SAME AS ABOVE FOR B'S                                   LOGI  66
COL  70 IY=NO OF STAGES BETWEEN COMPUTING ALL PARAMETERS
COL 75 ITONE = MAXIMUM NUMBER OF ITERATIONS FOR THETA
C                                                                      LOGI  70
      JSTAGE=0
      NKITE=0
      KITE(1)=0
      OLDSUM=0.
      KEXP=-99999999
      ALIM=10.E0
      CAPC=0.E0
      IF(CHOICE.NE.0.) CAPC=1./CHOICE
      INPUT=INPUT+1
      READ(5,2908)TIMAX,TOL1,CLAMDA,CBAR,CLO,CHI,TOL,CGUESS,AINIT,ALO,
     1AHI ,ACRIT,ALIM
 2908 FORMAT(F6.0,12F6.2)
      IF(NCH.EQ.100) CAPC=CBAR
      READ(5,1001) MAXOUT,ICKA,   NKILL,(KILL(M),M=1,NKILL)
C                                                                      LOGI 104
C      NKILL= NO. OF EXAMINEES TO BE SKIPPED                           LOGI 105
C       *** A MAXIMUM OF 99 PEOPLE MAY BE REMOVED - I.E. NKILL=99      LOGI 106
```

```
C      KILL(M) = NOS. THESE EXAMINEES WOULD HAVE HAD                    LOGI 107
C      MAXOLT= MAXIMUM NO. OF ITEMS WHICH WILL BE ALLOWED TO BE REMOVED LOGI 108
C      ICKA= 0 CHECK THE SIZE OF A AS A CRITERIAN FOR REMOVING ITEMS    LOGI 109
C      ICKA= 1 DO NOT CHECK A                                           LOGI 110
       IF(IT.LT.2) READ(5,1005)(ITEMNO(I),I=1,LITTLN)
 1005 FORMAT (2613,2X)                                                  LOGI 126
C      WRITE OUT INPUT OPTIONS
       IF(IT.LT. 2) GO TO 5500
       WRITE(6,5000)
       GO TO 5508
 5500 WRITE(6,5001)
       IF(IA.EQ.1) GO TO 5501
       WRITE(6,5002) AINIT
       GO TO 5502
 5501 WRITE(6,5003)
 5502 IF(IB.EQ.1) GO TO 5503
       WRITE(6,5004)
       GO TO 5504
 5503 WRITE(6,5005)
 5504 IF(NCH.NE.1000) GO TO 5505
       WRITE(6,5007)
       GO TO 5506
 5505 WRITE(6,5006) CAPC
 5506 IF(IT.EQ.1) GO TO 5507
       WRITE(6,5008)
       GO TO 5508
 5507 WRITE(6,5009)
 5508 WRITE(6,5010) MAXOUT
       IF(ICKA.EQ.0) GO TO 5509
       WRITE(6,5011)
       GO TO 5510
 5509 WRITE(6,5012) ALIM
 5510 WRITE(6,5013) IY,ACRIT
       WRITE(6,5014)
       IF(NC.EQ.2) GO TO 5521
       WRITE(6,5022) ALO,AHI
       GO TO 5522
 5521 WRITE(6,5015)
 5015 FORMAT(6X,'A''S WILL BE HELD FIXED')
       GO TO 5525
 5522 IF(NC.GE.0) GO TO 5523
       NCC=IABS(NC)
       WRITE(6,5016) NCC
 5016 FORMAT(6X,'C''S WILL BE HELD FIXED UNTIL STAGE ',I5)
       GO TO 5526
 5523 IF(NC.EQ.1 .OR. NCH .EQ.0) GO TO 5525
 5526 WRITE(6,5017) CLAMDA,CBAR,CLC,CHI
       GO TO 5524
 5525 WRITE(6,5018)
 5018 FORMAT(6X,'C''S WILL BE HELD FIXED')
```

```
5524 IF(NIT.EQ.0) GO TO 5511
     WRITE(6,5020)
     GO TO 5512
5511 WRITE(6,5019)
5512 WRITE(6,5021) MAXST,TIMAX,ITONE,TOL1,MAXIT
     IF(IWILL.EQ.1) GO TO 5513
     WRITE(6,5025)
     GO TO 5514
5513 WRITE(6,5026)
5514 IF(ILAST.GT.0) GO TO 5515
     WRITE(6,5027)
     GO TO 5516
5515 WRITE(6,5028) ILAST
5516 WRITE(6,5029) INPUTB(INPUT)
     IF(NGT.EQ.0 .AND. NSKIP.EQ.0) GO TO 5517
     WRITE(6,5030)NGT,NSKIP
     GO TO 5518
5517 WRITE(6,5031)
5518 IF(IT.NE.2) GO TO 5519
     WRITE(6,5032)
     GO TO 5520
5519 WRITE(6,5033)
5520 WRITE(6,5034) CGUESS
     IF(IT.LT. 1) GO TO 24
5000 FORMAT('0INITIAL ESTIMATES ARE FROM THE OUTPUT OF A PREVIOUS RUN')
5001 FORMAT('0INITIAL ESTIMATES WILL EITHER BE COMPUTED OR READ')
5002 FORMAT(6X,'INITIAL ESTIMATE FOR A IS ',F8.2)
5003 FORMAT(6X,'INITIAL ESTIMATES FOR A WILL BE READ')
5004 FORMAT(6X,'INITIAL ESTIMATES FOR B WILL BE COMPUTED')
5005 FORMAT(6X,'INITIAL ESTIMATES FOR B WILL BE READ')
5006 FORMAT(6X,'INITIAL ESTIMATE FOR C IS ',F8.2)
5007 FORMAT(6X,'INITIAL ESTIMATES FOR C WILL BE READ')
5008 FORMAT(6X,'INITIAL ESTIMATES FOR THETA WILL BE COMPUTED')
5009 FORMAT(6X,'INITIAL ESTIMATES FOR THETA WILL BE READ')
5010 FORMAT('0A MAXIMUM OF ',I4,' ITEMS MAY BE REMOVED')
5011 FORMAT(6X,'A''S ARE NOT TO BE USED AS A CRITERION FOR REMOVING ITE
    1MS')
5012 FORMAT(6X,'ITEMS WILL BE REMOVED IF THEIR A BECOMES GREATER THAN',
    1F8.2)
5013 FORMAT('CALL PARAMETERS WILL BE ESTIMATED EVERY',I4,' STAGES'/
    110X,'AND'/6X,'AFTER THE MAXIMUM CHANGE IN A BETWEEN TWO STAGES IS
    2LESS THAN ',F6.4)
5014 FORMAT('0THE FOLLOWING RESTRICTIONS ARE IMPOSED ON THE ITEM PARAME
    1TERS')
5022 FORMAT(6X,'A''S MUST BE WITHIN THE RANGE OF ',F8.2,' TO ',F8.2)
5017 FORMAT(
    26X,'C''S THE MAXIMUM ALLOWABLE CHANGE IN C IS ',F8.2/8X,'THE AVERA
    2GE VALUE FOR C IS ',F8.2/8X,'C IS RESTRICTED TO A RANGE OF ',F8.2,
    3' TO ',F8.2/10X,'UNLESS IT HAS A SMALL STANDARD ERROR')
5019 FORMAT('0THE RUN WILL BEGIN BY ESTIMATING ITEM PARAMETERS')
```

```
5020 FORMAT('0THE RUN WILL BEGIN BY ESTIMATING ABILITY PARAMETERS')
5021 FORMAT('0THE RUN WILL TERMINATE AFTER',I4,' STAGES OR AFTER', F8.0
    1,' SECONDS'/6X,'EACH THETA WILL BE ITERATED FOR',I4,' ITERATIONS O
    2R UNTIL THE CHANGE IN THETA IS LESS THAN ',F8.2/6X, 'THE MAXIMUM NO
    10 OF ITERATIONS PER ITEM IS ',I4)
5025 FORMAT('0PRINTOUT'/6X,'NO DETAILED PRINTOUT FOR THETAS')
5026 FORMAT('0PRINTOUT'/6X,'DETAILED PRINTOUT FOR THETAS')
5027 FORMAT(6X,'FIRST AND SECOND DERIVATIVES FOR ITEMS WONT BE PRINTED')
    1)
5028 FORMAT(6X,'FIRST AND SECOND DERIVATIVES FOR ITEMS WILL BE PRINTED
    1FOR ',I6,' STAGES')
5029 FORMAT('0ITEM RESPONSE INPUT IS ',A6)
5030 FORMAT(6X,'THE FIRST',I4,' RECORDS WILL BE SKIPPED, AND EVERY ',I4
    1,' RECORD  SELECTED THEREAFTER')
5031 FORMAT(6X,'ALL RECORDS WILL BE SELECTED')
5032 FORMAT('0THE INTERMEDIATE DATA SETS ARE ALREADY CREATED')
5033 FORMAT('0THE INTERMEDIATE DATA SETS WILL BE CREATED')
5034 FORMAT('0THE PROBABILITY OF GETTING AN ITEM CORRECT BY GUESSING AT
    1 RANDOM IS',F8.2)
     READ(5,7000)INP,JCHTH,MI(2),MI(3),MI(4),MI(5),MI(6),MI(7)
    1,MI(8),MI(9)
     IF(INP.EQ.0) GO TO 2660
7000 FORMAT(40I2)
      READ(5,534) FMT,MLITN,MLEST,MN,NEST,(QC(I),I=1,5),JSTAGE
     WRITE(6,2502) FMT,MLITN,MLEST,MN,NEST,(QC(I),I=1,5)
2502 FORMAT('0INPUT IS FROM RUN TITLED ',20X,10A6/10X,'NO. OF ITEMS REA
    1D ',I4,' NO OF ITEMS ESTIMATED ' ,
    1I4,'  NO. OF EXAMINEES READ IN = ',I8,'  NO. OF THETAS ESTIMATED =
    2',I8/10X,'RUN AT ',5A4)
     READ(5,2844)RPROD,LEXP,OLDSUM,KEXP,TOL
     GO TO 703
2660 READ(5,1002)FMT
1002 FORMAT(10A8)
     READ(5,FMT)(THETA(I),I=1,MN)
6012 IF(IT.EQ.1) GO TO 708
     DO 122 K=1,MLITN,6
     J=K+5
     IF(J.GT. MLITN) J=MLITN
     READ(5,123) (A(LL),B(LL),C(LL),LL=K,J)
 123 FORMAT(4X,18A4)
 122 CONTINUE
     READ(5,1001)(ITEST(I),I=1,MLITN)
     READ(5,1005)(ITEMNO(I),I=1,MLITN)
     READ(5,1001) NKILLM,(KILLS(M),M=1,NKILLM)
     READ(5,1001) NKITE,(KITE(M),M=1,NKITE)
     KITE(NKITE+1)=0
     GO TO 708
 703 DO 6010 I=1,MN,19
     READ(5,7001) THETT
     DO 6011 K=1,19
```

```
      L=I+K-1
      THETA(L)=THETT(K)
      IF(L.EQ.MN) GO TO 6012
 6011 CONTINUE
 6010 CONTINUE
 7001 FORMAT(4X,19A4)
  708 IF(JCHTH.EQ.0) GO TO 710
C     REPLACE THETAS, READ IN EXAMINEE NUMBER AND NEW THETAS--
C               MAXIMUM OF 200
      READ(5,7003)NOUR,(NNQ,THETA(NNQ),I=1,NOUR)
 7003 FORMAT(I5/(I5,E15.8,I5,E15.8,I5,E15.8,I5,E15.8))
  710 IF(IT.EQ.1) GO TO 24
      IA=1
      IB=1
      K=MI(2)+MI(3)+MI(4)+MI(5)+MI(6)+MI(7)+MI(8)+MI(9)
      IF(K.NE.0) CALL MODIFY(ITEST,THMED,Z,QCFR,KDC,NOM ,MI,MLITN)
  745 LITTLN=MLITN
   24 TOLA=TCL
      IF (KILL(1).EQ.0) GO TO 3003
      WRITE (6,3000) (KILL(M),M=1,NKILL)                              LOGI 120
 3000 FORMAT('0THE FOLLOWING EXAMINEES WILL BE SKIPPED'/
     1(' ',25I5))
 3003 CALL RESCLK(2,TIME)
      IF(NKITE.NE.0) WRITE(6,3005) (KITE(M),M=1,NKITE)
 3005 FORMAT('0 THE FOLLOWING ITEMS WILL BE SKIPPED '/(10X,20I5))
      KILL(NKILL+1)=0
      FLITN=LITTLN
      NLP=(N-1)/16+1
      NIPREC=752/NLP
      NRECPI=(LITTLN-1)/NIPREC+1
      NLB=(LITTLN-1)/16+1
      NPPREC=600/NLB
      NLEN=601-NLB
      DO 3207 K=1,LITTLN
      NPPI(K)=0
      NOM(K)=0
 3207 CONTINUE
 3601 JCOU=0
      IF(IT-2) 2501,3250,2503
 2503 OLDSUM=0.
      KEXP=-99999999
      IT=2
 2501 IF(IB.NE.0) GO TO 3200
      DO 63 K=1,LITTLN
   63 NB(K)=0
 3200 CALL OPEN(10,-11,-12)
   61 IF(NOT.EQ.0) GO TO 8102
      DO 8101 I=1,NOT
 8101 CALL GET(10,KDO)
 8102 JITSK= (LITTLN-1)/3+1
```

```
      KIND=1
      I=0
      IZ=1
      M=1
      KRE=KILL(1)
      DO 8505 II=1,N
      DO 8503 L=1,NSKIP
 8503 CALL GET(10,KDO)
      SC=0.
      JRE=0
      JSK=0
      IF(INPUT.EQ.2) GO TO 8530
C     INPUT IS BINARY
      K2=0
      DO 8506 KL=1,NLB
      K2=K2+16
      K1=K2-15
      IF(K2.GT. LITTLN) K2=LITTLN
      L=-2
      DO 8507 K=K1,K2
      L=L+2
      IK=ITEMNO(K)-1
      KWORD=IK/16
      KBIT=(IK-KWORD*16)*2
      KWORD=KWORD+1
      CALL MOV2BI(KBIT,KDO(KWORD),L,ZA)
      JANS=I2BIT(L,ZA)
      IF(JANS-2) 8514,8515,8507
 8514 JSK=JSK+1
      SC=SC+JANS
      GO TO 8516
 8515 SC=SC+CGUESS
 8516 JRE=JRE+1
 8507 CONTINUE
      Z(IZ)=ZA
 8506 IZ=IZ+1
 8538 IF(SC.EQ.0.) GO TO 8509
      IF(SC.EQ. FLITN) GO TO 8509
      IF(JSK.LE. JITSK) GO TO 8508
      IF(II.EQ.KRE) GO TO 8501
      IF(IT.EQ.0) THETA(II)=SC/JRE
C      FIRST RUN ON A SET OF DATA- COMPUTE NB(K) AND NPPI(K)
      JZ=IZ-NLB
      K2=0
      DO 3210 KL=1,NLB
      K2=K2+16
      K1=K2-15
      IF(K2.GT. LITTLN) K2=LITTLN
      L=-2
      ZA=Z(JZ)
```

```
      DO 3211 K=K1,K2
      L=L+2
      IJ=I2BIT(L,ZA)
      IF(IJ.LT.2 .AND. IB.EQ.0) NB(K)=NB(K)+IJ
      IF(IJ.LT.3) NPPI(K)=NPPI(K)+1
      IF(IJ.EQ.2) NOM(K)=NOM(K)+1
 3211 CONTINUE
 3210 JZ=JZ+1
      GO TO 8510
C     INPUT IS BCD
 8530 K2=0
      DO 8536 KL=1,NLB
      K2=K2+16
      K1=K2-15
      IF(K2.GT. LITTLN) K2=LITTLN
      L=-2
      DO 8537 K=K1,K2
      L=L+2
      JJ=(ITEMNO(K)-1)*2+1
      J=ICOLS(JJ,JJ+1)
C      NOTE THAT THE LAST TWO BITS OF 10 AND 11 INTEGER ARE 10 AND 11
C             BINARY
      CALL MOV2BI(30,J,L,ZA)
      IF(J-10) 8513,8512,8537
 8513 JSK=JSK+1
      SC=SC+J
      GO TO 8518
 8512 SC=SC+CGUESS
 8518 JRE=JRE+1
 8537 CONTINUE
      Z(IZ)=ZA
 8536 IZ=IZ+1
      GO TO 8538
 8509 WRITE(6,2890) II,JSC
 2890 FORMAT('0EXAMINEE ',I6,' HAD A SCORE OF ',I4,' AND IS BEING REMOVE
     1D')
      GO TO 8504
 8501 M=M+1
      KRE=KILL(M)
 8508 WRITE(6,2891)II
 2891 FORMAT('0EXAMINEE ',I6,' HAS BEEN REMOVED ON REQUEST OR BECAUSE HE
     1 DID NOT ANSWER OVER A THIRD OF THE ITEMS')
 8504 I=I+1
      IF(I.GT. 100) GO TO 4400
      KILLS(I)=II
      THETA(II)=-999999.
 8510 IF(IZ.LE.NLEN) GO TO 8505
      CALL PUT(11,Z)
      IZ=1
 8505 CONTINUE
```

```
      IF(IZ.NE.1) CALL PUT(11,Z)
      NKILL=I
      KILLS(NKILL+1)=C
      TOTAL = N-NKILL
      TOTSQ=TOTAL*TOTAL
C     WRITE THE ITEM RESPONSES BY ITEM ON UNIT 12
      NPREVC=601-NLB
      JBEGA=1
 3221 CALL CLOSE(11)
      CALL REOPEN(11)
      KWEN=5
      KSTO=JBEGA+5*NIPREC
      IF(KSTO.LE. LITTLN) GO TO 3227
      KWEN=(LITTLN-JBEGA)/NIPREC+1
 3227 IZ=1
      CALL GET(11,Z)
      IF(JBEGA.NE.1) GO TO 4301
      IEND=NLB*10
      DO 935  I=1,IEND,NLB
      JK=I+NLB-1
      WRITE (6,931) (Z(L),L=I,JK)
  931 FORMAT(' ',6Z8)
  935 CONTINUE
 4301 DO 3222 II=1,N
C     POSITION IN ITEM VECTOR OF RESPONSES
      IK=II-1
      IWORD=IK/16
      MBEG=IWORD+1
      IBIT=(IK-16*IWORD)*2
      DO 3223 J=1,KWEN
      LBEG=MBEG
      JBEG=(J-1)*NIPREC+JBEGA
      JEND=JBEG+NIPREC-1
      IF(JEND.GT. LITTLN) JEND=LITTLN
      DO 3224 K=JBEG,JEND
      JZ=K-1
      IK=JZ/16
      KBIT=(JZ-16*IK)*2
      CALL MOV2BI(KBIT,Z(IZ+IK),IBIT,ZZ(LBEG))
      LBEG=LBEG+NLP
 3224 CONTINUE
      MBEG=MBEG+752
 3223 CONTINUE
      IZ=IZ+NLB
      IF(IZ.LE. NPREVC) GO TO 3222
      IZ=1
      CALL GET(11,Z)
 3222 CONTINUE
      DO 3225 K=1,KWEN
      KBEG=(K-1)*752+1
```

```
      CALL PUT(12,ZZ(KBEG))
 3225 CONTINUE
      IF(JEND.EQ. LITTLN) GO TO 3226
      JBEGA=KSTO
      GO TO 3221
 3226 CALL CLOSE(11)
      CALL CLOSE(12)
      CALL REOPEN(11)
      CALL REOPEN(12)
      CALL CLOSE (10)
   10 IC=3
      IF(IT.EQ.2) GO TO 520
      IF (NC.EQ.1) IC=2                                          LOGI  96
      IF(NC.LT.0) IC=2
      IF(NC.EQ.2) IC=1
      IF(NCH.EQ.0) IC=2
      DO 3100 K=1,LITTLN                                         LOGI 100
 3100 KDO(K)=IC                                                  LOGI 101
C                                                                LOGI 198
C                    COMPUTE INITIAL ESTIMATES                   LOGI 199
C                                                                LOGI 200
      CALL RESCLK (2,TIME)                                       LOGI 204
      IF(IT.NE.0 .AND. IB.NE.0) GO TO 26
      SUM=0.
      SUMSQ=0.
      M=1
      KRI=KILLS(1)
      DO 20 I=1,N                                                LOGI 208
      IF(I.EQ. KRI) GO TO 21
      T1=THETA(I)
      SUM=SUM+T1
      SUMSQ=SUMSQ+T1*T1
      GO TO 20
   21 M=M+1
      KRI=KILLS(M)
   20 CONTINUE                                                   LOGI 221
      XMEAN=SUM/TOTAL
      VAR=TOTAL*SUMSQ-SUM*SUM
      VAR=VAR/TOTSQ                                              LOGI 224
      SD= SQRT(VAR)
   23 WRITE(6,9006) XMEAN,VAR
 9006 FORMAT('0NUMBER RIGHT SCORES ADJUSTED FOR OMITS ', 10X,'MEAN = ',
     1F9.5,2X,'VARIANCE = ',F11.5)
      IF(IT.NE. 0)  GO TO 26
 9001 FORMAT(' ',I5,3X,20I5)
      WRITE(6,2897)
 2897 FORMAT(20X,'SCORES FOR REMOVED EXIMINEES WILL BE PRINTED AS ASTERI
     2SKS')
      M=1
      KRI=KILLS(1)
```

```
      DO 261 J=1,N,25
      JK=J+24
      IF(JK.GT.N) JK=N
      WRITE(6,2030)J,(THETA(L),L=J,JK)
      DO 2861 KK=J,JK
      IF(KK.EQ.KRI) GO TO 2862
      THETA(KK)=(THETA(KK)-XMEAN)/SD
      GO TO 2861
 2862 M=M+1
      KRI=KILLS(M)
 2861 CONTINUE
  261 CONTINUE
      GO TO 26
  520 DO 522 K=1,LITTLN
      KDO(K)=ITEST(K)
  522 CONTINUE
   26 NEX=N-NKILL
      MLITN=LITTLN-NKITE
      WRITE(6,2000) TITLE,LITTLN,MLITN,CATER,N,NEX
      TOTAL=NEX
      TOTSQ=TOTAL*TOTAL
      IF(IT.EQ.2) GO TO 36
      M =1
      KRE=KITE(1)
      DO 30 K=1,LITTLN                                                    LOGI 271
      IF(ITEMNO(K) .EQ. KRE) GO TO 3888
      A(K)=AINIT
      IF(IB.EQ.0) B(K)=(FLOAT(N8(K))+CGUESS*NOM(K))/FLOAT(NPPI(K))
      C(K)=CAPC
      GO TO 30
 3888 M =M +1
      KRE=KITE(M )
   30 CONTINUE                                                            LOGI 277
      IF(IA.EQ.0) GO TO 33
      READ (5,1002) FMT                                                   LOGI 286
      READ (5, FMT) (A(K),K=1,LITTLN)                                     LOGI 287
   33 IF (IB.EQ.0) GO TO 34                                               LOGI 288
      READ (5,1002) FMT                                                   LOGI 289
      READ (5, FMT) (B(K),K=1,LITTLN)                                     LOGI 290
      GO TO 31
   34 WRITE(6,9004)
 9004 FORMAT('1PROPORTION OF CORRECT RESPONSES ADJUSTED FOR GMITS'/
     1'0    K    ITEM(K)',14X,'P')
 9003 FORMAT (1H ,I4,I10,F20.8)                                           LOGI 283
      M =1
      KRE=KITE(1)
      DO 35 K=1,LITTLN
      IF(ITEMNO(K) .EQ. KRE) GO TO 3588
       WRITE(6,9003) K,ITEMNO(K),B(K)
      V1=1.EO-B(K)
```

```
      B(K)=SQRT(1.+A(K)*A(K))*PINV(V1)/A(K)
      GO TO 35
 3588 M =M +1
      KRE=KITE(M )
   35 CONTINUE
   31 IF(NCH.EQ.1000) READ(5,1004) (C(K),K=1,LITTLN)
 1004 FORMAT (8F10.0)                                                  LOGI 303
   36 WRITE (6,9005)                                                   LOGI 294
 9005 FORMAT('0  INITIAL ESTIMATES')
      IF(KIND.EQ.0) GO TO 64
      WRITE(6,9021)
      M=1
      KRE=KITE(1)
      DO 610 K=1,LITTLN
      IK=ITEMNO(K)
      IF(IK.EQ. KRE) GO TO 611
       WRITE(6,9022) IK,A(K),B(K),C(K),KDO(K),NPPI(K),NOM(K)
      GO TO 610
  611 M=M+1
      KRE=KITE(M)
  610 CONTINUE
      GO TO 62
   64 WRITE(6,9023)
      M=1
      KRE=KITE(1)
      DO 612 K=1,LITTLN
      IK=ITEMNO(K)
      IF(IK.EQ. KRE) GO TO 613
      WRITE(6,9024) IK,A(K),B(K),C(K),KDO(K)
      GO TO 612
  613 M=M+1
      KRE=KITE(M)
  612 CONTINUE
   62 WRITE(6,9002)
 9002 FORMAT('0INITIAL  ESTIMATES OF THETAS',20X,'THETAS FOR REMOVED EXA
     1MINEES, THETAS .LE. -10, AND THETAS .GE. 100'/70X,'WILL BE PRINTED
     2 AS ASTERISKS')
      DO 266 I=1,N,25
      J=I+24
      IF(J.GT.N)J=N
      WRITE(6,2030) I,(THETA(L),L=I,J)
  266 CONTINUE
 2030 FORMAT(' ',I4,2X,25F5.2)
      ISTAGE=JSTAGE
      MAXST=JSTAGE+MAXST
   39 CALL RESCLK(2,TIME)
      IF(JCHTH.NE.0 .OR. IT.NE.2)  GO TO 50
      IF(NOCHEC.EQ.1) GO TO 50
      CALL PRODUC(PROD,IEXP)
 2849 IF(PROD.NE. RPROD) GO TO 2851
```

```
      IF(IEXP.NE. LEXP) GO TO 2851
   50 RETURN
 2851 WRITE(6,2852) PROD,RPROD,PRCD,RPRCD,IEXP,LEXP
 2852 FORMAT('0PRODUCT OF THETAS TO BE USED DOES NOT AGREE WITH PRODUCT
     1OF FINAL THETAS OF PREVIOUS RUN'/' ',Z16,1X,Z16,2024.16,2I6)
      GO TO 990
 4400 WRITE(6,4401)
 4401 FORMAT('0MORE THAN 100 PEOPLE HAVE BEEN REMOVED')
      CALL EXIT
 3250 CALL OPEN(11,12)
      NOCHEC=0
      IF(NKILL.EQ.0) GO TO 553
      IF(NKILLM.NE.0) GO TO 547
      DO 552 J=1,NKILL
      KJ=KILL(J)
      THETA(KJ)=-999999.
      KILLS(J)=KJ
  552 CONTINUE
      NOCHEC=1
      GO TO 520
  553 NKILL=NKILLM
      GO TO 520
  547 L=0
      K=1
      J=1
  545 KK=KILL(K)
  549 IF(KK-KILLS(J)) 541,542,543
  541 L=L+1
      NUM(L)=KK
  544 K=K+1
      IF(K.LE. NKILL) GO TO 545
      DO 546 JJ=J,NKILLM
      L=L+1
      NUM(L)=KILLS(JJ)
  546 CONTINUE
      GO TO 548
  543 L=L+1
      NUM(L)=KILLS(J)
      J=J+1
      IF(J.LE. NKILLM) GO TO 549
      GO TO 554
  542 J=J+1
      L=L+1
      NUM(L)=KK
      IF(J.LE. NKILLM) GO TO 544
      K=K+1
      IF(K.GT. NKILL) GO TO 548
  554 DO 550 JJ=K,NKILL
      L=L+1
      NUM(L)=KILL(JJ)
```

```
  550 CONTINUE
  548 DO 551 J=1,L
      KJ=NCM(J)
      THETA(KJ)=-999999.
      KILLS(J)=KJ
  551 CONTINUE
      NKILL=L
      NOCHEC=1
      GO TO 520
  990 CALL EXIT
 2844 FORMAT(A8,4A4)
  534 FORMAT(10A6,2I3,2I6/5A4,4I5)
 9021 FORMAT('0','ITEM(K)',15X,'A',19X,'B',19X,'C',9X,'PARAM CODE NO.PEO
     1PLE NO. OMITS')
 9023 FORMAT('0','ITEM(K)',15X,'A',19X,'B',19X,'C',9X,'PARAM CODE')
 9022 FORMAT(' ',I7,2F20.8,14X,F6.4,6X,I4,6X,I4,6X,I4,7X,I4)
 9024 FORMAT(' ',I7,2F20.8,14X,F6.4,6X,I4,6X,I4)
      END
      SUBROUTINE SUB2
C     ITERATES ON PARAMETERS SELECTED ON SIZE OF LAST CHANGE
C     ACTUAL SECOND DERIVATIVES USED-- NOT EXPECTED VALUES
      COMMON AHI,ALIM,ALO,ACRIT  ,CBAR,CGUESS,CHI,CHOICE,CLAMDA,CLO,
     1ICKA,ILAST,INDIC,ISTAGE,          JSTAGE,LITTLN,MAXOUT,MAXST,
     5MAXIT,IWILL,ITONE,NOUT,NRECPI,CLDSUM,KEXP,
     2N,NC,NIPREC,NIT,NKILL,NLB,NLP,NPPREC,    TIMAX,    TOL1,TOLA,
     3TOTAL,A(100),B(100),C(100),KILLS(100),KOO(100),ITEMNO(100),
     4IY,NKITE,KITE(100),TITLE(15),DATER(5),THETA(3000)
      DIMENSION VV(260),JSA(260)
      DIMENSION  Z(752),QCSQ(100),QC(100),QCFR(100),ITEST(100),
     1KILL(100),ADJ(6),V(13),JS(15),HOLL(3),TIMEC(5),WHY(5),CONVR(5)
     2,THETT(19),NCM(100),NPPI(100)
      REAL*8 PRODP,PRODQ,F122,F121
      REAL*8 U1,U2,U3,F11,F13,F22,F23,F33
      REAL*8 DT1,SSB,SSC
      REAL*8 A11,A12,A13,A22,A23,A33,SS1,SS2,SS3
      REAL*8 PROD,RPROD
      DIMENSION RR(11)
      EQUIVALENCE (RR(3),RDA),(RR(4),RA),(RR(5),VA),(RR(6),RDB),
     1(RR(7),RB),(RR(8),VB),(RR(9),RDC),(RR(10),RC),(RR(11),VC)
      EQUIVALENCE (V(1),SUMS),(V(2),PRODS),(V(12),TOL),(V(13),THCH),
     1(JS(1),JEXP),(JS(2),IEXP),(JS(12),JDRTH),(JS(13),JCHTH)
      REAL*4 INDC
      REAL*4 INFA
      DIMENSION UA(100),UB(100),UC(100)
      DIMENSION DTHET(3000),DA(100),DB(100),DC(100)
      DATA HOLL(1)/'A*'/,HOLL(2)/'B'/,HOLL(3)/'C'/
      EQUIVALENCE (ADJ(1),ADJ1),(ADJ(2),ADJ2),(ADJ(3),ADJ3),
     1(ADJ(4),ADJ4),(ADJ(5),ADJ5),(ADJ(6),ADJ6)
      DATA TIMEC/18H - OUT OF TIME  **/,CONVR/18H - MINIMUM FOUND */    LOGI   9
      DATA STAR/'*'/,STARR/'****'/,BLANK/' '/
```

```
   39 DO 41 L=1,5
   41 WHY(L)=BLANK                                                  LOGI 323
      NCH=IABS(NC)
      IF(IY.EQ.0) IY=1
      OMCG=1.-CGUESS
      IHAVE=0                                                       LOGI  87
      DO 291 K=1,LITTLN
      UA(K)=A(K)
      UB(K)=B(K)
      UC(K)=C(K)
  291 CONTINUE
      KSTAGE=ISTAGE +1
      IQUIT=1
      JEXP=0
      SUMS=0.
      JSUMC=0
      TIMTOT=0.
      NOUT=0
C                                                                   LOGI 329
C                 FIND THETA FOR EACH EXAMINEE                      LOGI 330
C                                                                   LOGI 331
   50 ISTAGE=ISTAGE+1                                               LOGI 332
      IF(ISTAGE.NE.NCH) GO TO 504
      IF(NC.GE.0) GO TO 504
      IFIRST=0
      DO 505 J=1,LITTLN
      KDO(J)=3
  505 CONTINUE
  504 IF(KSTAGE.NE. ISTAGE) GO TO 51
      IFIRST=0
      JFIRST=0
      KSTAGE=KSTAGE+IY
   51 TOL=0.
      STIME=CLOCK(ITIME)                                            LOGI 334
      ITMAXT=1                                                      LOGI 345
      THCH=0.
      JORTH=0
      FMAXT=0.
      JTHMAX=0
      NESTH=0
      JC=0
      JCHTH=0
      IF(NIT.EQ.0) GO TO 289
      JORTH=1
      JCHTH=1
C                                                                   LOGI 336
C               SET UP VECTORS INDEPENDENT OF THETA                 LOGI 337
C                                                                   LOGI 338
      M=1
      KRI=KILLS(1)
```

```
      DO 90  K=1,LITTLN                                                 LOGI 339
      QC(K)=1.E0-C(K)
      QCSQ(K)=A(K)/QC(K)                                                LOGI 341
      QCFR(K)=QCSQ(K)*QCSQ(K)                                           LOGI 342
   90 CONTINUE                                                          LOGI 343
      IEXP=0
       PRODP=1.D0
      PRODQ=1.D0
      JEXPP=0
      JEXPQ=0
      IT=1                                                              LOGI 347
      PROD=1.D0
      NPREVC=(N-1)/NPPREC+1
      NPRE1=NPPREC-1
      DO 200 IL=1,NPREVC
      CALL GET(11,Z)
      IZ=0
      J2=IL*NPPREC
      J1=J2-NPRE1
      IF(J2.GT.N) J2=N
      DO 200 I=J1,J2
      IF(I.EQ. KRI) GO TO 3010
      IF(IFIRST.NE.1) GO TO 600
      IF(DTHET(I).LT. FMAXTH) GO TO 204
  600 ITER=0
      NESTH=NESTH+1
  100  INDC=BLANK
C                                                                       LOGI 352
C               START OF ITERATIONS TO FIND THETA FOR ONE EXAMINEE      LOGI 353
C                                                                       LOGI 354
      THET=THETA(I)
      THETK=THET
      ITER=ITER+1
      INFA=0.E0
      U=0.E0
      DU=0.
      K2=0
      MI=1
      KRE=KITE(1)
 3600 DO 150 KL=1,NLB
      ZZ=Z(IZ+KL)
      K2=K2+16
      K1=K2-15
      IF(K2.GT. LITTLN) K2=LITTLN
      L=-2
      DO 150 K=K1,K2
      L=L+2
      IF(ITEMNO(K).EQ. KRE) GO TO 3024
      CK=C(K)
      JI=I2BIT(L,ZZ)
```

```
      IF(JI.EQ.3) GO TO 150
      T3=JI
      IF(JI.EQ.2) T3=CGUESS
      PSIARG=-1.702E0*A(K)*(-B(K)+THET)
      IF(PSIARG.LT. 19.E0) GO TO 120
      P=CK
      IF(P.EQ.0.) P=.0000001
      Q=1.-P
      GO TO 140
  120 P=CK+QC(K)/(1.E0+ EXP(PSIARG))
      IF(P.GT. .9999999) P=.9999999
      IF(P.LT. .0000001) P=.0000001
      Q=1.E0-P
      PC=1.EC-(CK/P)
      U=U+PC*(T3-P)*QCSQ(K)
      INFA=INFA+QCFR(K)*PC*Q*(P-(T3*CK/P))
      GO TO 140
 3024 MI=MI+1
      KRE=KITE(MI)
      GO TO 150
 3256 PRODP=PRODP*P
      IF(PRODP.GT. 1.D-35) GO TO 3200
      PRODP=PRODP*1.D+35
      JEXPP=JEXPP+1
 3200 PRODQ=PRODQ*Q
      IF(PRODQ.GT. 1.D-35) GO TO 150
      PRODQ=PRODQ*1.D+35
      JEXPQ=JEXPQ+1
      GO TO 150
C                                                                    LOGI 380
C              COMPUTE CRITERION  FUNCTION FOR CURRENT VALUES
C                                                                    LOGI 382
  140 IF(IFIRST.EQ. 1) GO TO 150
      IF(ITER.NE.1) GO TO 150
      IF(JI.EQ.2) GO TO 3256
      WHICH=Q
      IF(JI.EQ. 1) WHICH = P
 3257 PROD=PROD*WHICH
      IF(PROD.GT. 1.D-35) GO TO 150
      PROD=PROD*1.D+35
      IEXP=IEXP+1
  150 CONTINUE                                                       LOGI 397
 3241 IF(INFA.NE. 0.) GO TO 151
      DELTA=C.EO
      AK=C.EC
      GO TO 170
  151 IF(INFA.GT.0.) GO TO  3100
      IF(ITER.EQ.1) JC=JC+1
      INFA=-INFA
      INDC=STARR
```

```
3100 DELTA=U/(1.702*INFA)
     T1=ABS(U)
     IF(TOL.GT. T1) GO TO 3013
     TOL=T1
     JDRTH=I
3013 AK=ABS(DELTA)
     IF(AK.LT. 1.E0) GO TO 155
     IF(THET*(THET+DELTA).GT. 0.E0) GO TO 155
     THET=.5E0*THET- SIGN(1.E0,THET)
     WRITE(6,2190) I ,THET,DELTA,INDC
     DELTA=THET-THETK
     AK=ABS(DELTA)
     GO TO 195
3010 M=M+1
     KRI=KILLS(M)
     GO TO 204
 155 THET     =THET     +DELTA                                          LOGI 408
9000 FORMAT(' EXAMINEE ',I5,' ITERATION ',I3,' DELTA = ',E16.8,
    1' THETA = ',E16.8,' S= ',E16.8,' INFO = ',E16.8,1X,A4)
 170 TLAMDA= ABS(.3E0*THETK)+2.E0
     IF(AK.LT. TLAMDA) GO TO 195
     AK=TLAMDA
     IF(DELTA.LT.0.) TLAMDA=-TLAMDA
     DELTA=TLAMDA
     THET=THETK+DELTA
     WRITE (6,2174)                                                    LOGI 435
2174 FORMAT (1H ,5H*****,20X,14HTHETA ADJUSTED,20X,5H*****  )          LOGI 436
     WRITE(6,9000) I ,ITER,DELTA,THET,U,INFA ,INDC
 195 IF(IWILL .EQ.1 .OR. INDC.EQ. STARR) WRITE(6,9000) I,ITER,DELTA,
    1THET,U,INFA,INDC
     THETA(I)=THET
     IF(ABS(THET).GT.2.) GO TO 601
     IF(AK.LT. ABS(FMAXT)) GO TO 601
     FMAXT=DELTA
     JTHMAX=I
 601 DTHET(I)=AK
     IF(AK.LT. ABS(THCH)) GO TO 902
     THCH=DELTA
     JCHTH=I
 902 IF(ITONE.LE.1 ) GO TO 204
     IF(ITER.GT. ITMAXT) ITMAXT=ITER
     IF(ITER.GE. ITONE) GO TO 204
C                                                                      LOGI 455
     IF(AK.GT. TOL1) GO TO 100
 204 IZ=IZ+NLB
 200 CONTINUE                                                          LOGI 456
     LOG=1
     IF(IFIRST.EQ.1) WRITE(6,620) FMAXTH
 620 FORMAT('  THETAS WERE NOT ESTIMATED IF THEIR LAST CHANGE WAS LESS
    1THAN ',F16.8)
```

```
      WRITE(6,621)NESTH
  621 FORMAT(' ',I6,'  THETAS WERE ESTIMATED')
      WRITE(6,602) FMAXT,JTHMAX
  602 FORMAT('0MAXIMUM CHANGE FOR THETAS IN THE RANGE OF -2 TO +2 IS ',
     1F14.8,' FOR EXAMINEE ',I4)
      FMAXTH=.2*ABS(FMAXT)
      IF(JC.NE.0) WRITE(6,3101)JC
 3101  FORMAT('0',10X,I6,' SECOND DERIVATIVES FOR THETAS WERE NEGATIVE')
      IF(IFIRST.EQ.1) GO TO 4401
 3207 PRODQ=PRODQ**OMCG
      PRODP=PRODP**CGUESS
      PROD=PROD*PRODP
      IF(PROD.GE.1.D-35) GO TO 3201
      PROD=PROD*1.D+35
      IEXP=IEXP+1
 3201 PROD=PROD*PRODQ
      IF(PROD.GE. 1.D-35) GO TO 3202
      PROD=PROD*1.D+35
      IEXP=IEXP+1
 3202 T1=FLOAT((JEXPP-JEXPQ)*(-35))*CGUESS
      IEXP=IEXP+JEXPQ
      JK=T1
      T2=T1-JK
      PROD=PROD*(10.**T2)
      IF(PROD.GE. 1.D-35) GO TO 3203
      PROD=PROD*1.D+35
      IEXP=IEXP+1
 3203 IEXP=(IEXP*(-35))+JK
 3204 IF(PROD.GE. .100) GO TO 3205
      PROD=PROD*10.
      IEXP=IEXP-1
      GO TO 3204
 3205 IF(LOG.EQ.2) GO TO 3206
      J=ISTAGE-1
      WRITE(6,2020) J,PROD,IEXP
 2020 FORMAT('0*****  CRITERION  AFTER CHANGE IN ITEM PARAMETERS IN STAG
     1E ',I3,' *****  L = ',F15.8,'*10**(',I10,')')
C                                                                       LOGI 459
C               TEST FOR CONVERGENCE                                    LOGI 460
C                                                                       LOGI 461
      IF(JFIRST.EQ.1 .OR. INDCC.EQ.1) GO TO 701
      IF(OLDSUM.GE. PROD) GO TO 901
      IF (KEXP.NE.IEXP) GO TO 901                                       LOGI 463
  700 WRITE(6,2700)
 2700 FORMAT (1HO,15HL HAS INCREASED,5X,1H*  )                          LOGI 467
  701 WRITE(6,2001) ISTAGE
 2001 FORMAT (1H1,18HBEGINNING OF STAGE,I3)                             LOGI 474
C  NORMALIZE UNGROUPED THETAS
      OLDSUM=PROD
      KEXP=IEXP
```

```
4401 IP23N=0
     JEXP=IEXP
     SUMS=PROD
     A12=0.DO
     A11=0.DO
     IP23N=0
     M=1
     KRI=KILLS(1)
     DO 250 I=1,N
     IF(I.EQ. KRI) GO TO 2502
     TH=THETA(I)
     IF(ABS(TH).GT. 10.) GO TO 250
     IP23N=IP23N+1
     A11=A11+TH
     A12=A12+TH*TH
     GO TO 250
2502 M=M+1
     KRI=KILLS(M)
 250 CONTINUE
     A13=IP23N
     A22=A11/A13
     A33=A12-A11*A22
     IF(A33.LE.0.DO) GO TO 259
     X=DSQRT(A33/(A13-1.DO))
     X23=A22
     WRITE(6,2285)X23,X,IP23N
2285 FORMAT('0 THE MEAN = ',F11.8,'  THE STD. DEV. = ',F11.8,' FOR THE
    1',I6,' THETAS BETWEEN -10 AND +10')
     GO TO 257
 259 WRITE(6,2259)
2259 FORMAT('0VARIANCE OF THE THETAS IS LESS THAN OR EQUAL TO ZERO')
     GO TO 900
 901 IF (IEXP.GT.KEXP) GO TO 700
     IQUIT=-1
     GO TO 701                                                          LOGI 916
2190 FORMAT (1H0,5H*****,15X,6HTHETA(,I4,17H) WILL BE SET TO  ,F13.8,   LOGI 441
    120X,5H*****  ,10X,'DELTA = ',D16.8,4X,A4)
 257 IF(NC.EQ.2) X=1.
     XINV=1./X
     WRITE(6,302)ISTAGE
 302 FORMAT('0   THETAS WITH ABSOLUTE VALUE GREATER THAN 3 IN STAGE ',
    1I5)
     J=0
     M=1
     KRI=KILLS(1)
     DO 258 K=1, N
     IF(K.EQ.KRI) GO TO 2858
     TH=(THETA(K)-X23)*XINV
     IF( ABS(TH).LT.3.EQ) GO TO 304
     J=J+1
```

```
      ITEST(J)=K
      QC(J)=TH
      IF(J.LT.7) GO TO 304
      WRITE(6,303) (ITEST(L),QC(L),L=1,J)
  303 FORMAT(' ',7(I8,F10.4))
      J=0
  304 THETA(K)=TH
      GO TO 258
 2858 M=M+1
      KRI=KILLS(M)
  258 CONTINUE
      IF(J.NE.0) WRITE(6,303) (ITEST(L),QC(L),L=1,J)
       WRITE(6,7653)
      M=1
      KRE=KITE(1)
      DO 265 K=1,LITTLN
      IF(ITEMNO(K).EQ. KRE) GO TO 2650
      B(K)=(B(K)-X23)*XINV
      A(K)=A(K)*X
      IF(A(K).GT.AHI) A(K)=AHI
      WRITE(6,7654)ITEMNO(K),A(K),B(K)
      GO TO 265
 2650 M=M+1
      KRE=KITE(M)
  265 CONTINUE
 1262 CALL RESCLK(2,TIME)
  289 PROD=1.D0
      JS(14)=NESTH
      INDCC=0
      PRODP=1.D0
      PRODQ=1.D0
      NESTH=0
      JEXPP=0
      JEXPQ=0
      IEXP=0
C                                                                      LOGI 539
C        FIND PARAMETERS FOR EACH ITEM                                 LOGI 540
C                                                                      LOGI 541
      ITMAX=1
      KEEP=0                                                           LOGI 543
      DO 4005 JJ=3,11
      V(JJ)=0.
      JS(JJ)=1
 4005 CONTINUE
      WRITE(6,2608)ISTAGE,TOLA
 2608 FORMAT('1 STAGE  ',I4/'0 TOLERANCE FOR CONVERGENCE OF A IS = ',
     1F12.4)
      IF(IFIRST.EQ.1) WRITE(6,623)DAA,DBB,DCC
  623 FORMAT('  ITEMS WILL BE SKIPPED IF THE LAST CHANGES IN A, B, AND C
     1 WERE LESS THAN ',3F9.5,' RESPECTIVELY')
```

```
      WRITE(6,2400)
 2400 FORMAT('0ITEM ITER NO',6X,'DERIVATIVES',15X,'A',7X,'DELTA',7X,'B',
     1 7X,'DELTA',7X,'C',6X,'DELTA',5X,'STD. DEV.',4X,'FIXED'/9X,'PARAM'
     3,' A',
     28X,'B',8X,'C',21X,'A',19X,'B',18X,'C',11X,'C')
      ICA=0
      MI=1
      KRE=KITE(1)
      DO 500 KL=1,NRECPI
      CALL GET(12,Z)
      LBEG=0
      K2=KL*NIPREC
      K1=K2-NIPREC+1
      IF(K2.GT. LITTLN) K2=LITTLN
      DO 500 K=K1,K2
      IK=ITEMNO(K)
      IF(IK.EQ. KRE) GO TO 5001
      ID=KDO(K)
      IF(IFIRST .NE. 1) GO TO 604
      IF(IL.LT.0) GO TO 604
      IF(DA(K).GT. DAA) GO TO 604
      IF(DB(K).GT. DBB) GO TO 604
      IF(DC(K).GT. DCC) GO TO 604
C     SKIP THIS ITEM
      LBEG=LBEG+NLP
      GO TO 500
  604 DO 4000 JJ=3,11
      RR(JJ)=0.
 4000 CONTINUE
      SA=0.
      SB=0.
      SC=0.
      NESTH=NESTH+1
      NPP=0
      NO=0
      IC=IABS(ID)
      IF(IC.GT.10) IC=IC-10
      ICA=IC
      ITER=0                                                        LOGI 554
      JLI=0
C                                                                   LOGI 557
C               START OF ITERATIONS TO FIND PARAMETERS FOR ONE ITEM LOGI 558
C                                                                   LOGI 559
      AK=A(K)
      BK=B(K)
      CK=C(K)
      WRITE(6,3004) IK,AK,BK,CK
 3004 FORMAT('0',I4,33X,F9.4,13X,F8.4,13X,F7.4)
      OLDA=AK
      OLDB=BK
```

```
      OLDCC=CK
  301 ITER=ITER+1
      SCC=0.
      NOAD=0                                                    LOGI 561
      DO 290 L=1,6
  290 ADJ(L)=BLANK                                              LOGI 563
      SAVEA=AK
      SAVEB=BK
      SAVEC=CK
      U1=0.DO
      U2=0.DO
      U3=0.DO
      F11=0.DO
      F121=0.DO
      F122=0.DO
      F13=0.DO
      F22=0.DO
      F23=0.DO
      F33=0.DO
      QD=1.E0-CK
      QCI=1.E0/QD
      M=1
      KRI=KILLS(1)
      I2=0
      DO 400 II=1,NLP
      LBEG=LBEG+1
      I2=I2+16
      I1=I2-15
      IF(I2.GT. N) I2=N
      ZZ=Z(LBEG)
      L=-2
      DO 400 I=I1,I2
      L=L+2
      IF(I.EQ. KRI) GO TO 401
      JRIQ1=I2BIT(L,ZZ)
      IF(JRIQ1.EQ.3) GO TO 400
      RIQ1=JRIQ1
      IF(JRIQ1.EQ.2) RIQ1=CGUESS
      TB=-BK+THETA(I)
      PSIARG=-1.7C2E0*AK*TB
      IF(ABS(PSIARG) .GT. 170.) GO TO 206
      P=CK+QD/(1.E0+ EXP(PSIARG))
      IF(P.GT. .9999999) P=.9999999
  202 IF(P.LT. .0000001) P=.0000001
      Q=1.E0-P
      IF(JLI.NE.0) GO TO 630
      NPP=NPP+1
      IF(JRIQ1.LE.1) GO TO 3262
      NO=NO+1
      IF(IFIRST.EQ.1) GO TO 630
```

```
      PRODP=PROLP*P
      IF(PRODP.GT. 1.D-35) GO TO 3210
      PRODP=PRODP*1.D+35
      JEXPP=JEXPP+1
 3210 PRODQ=PRODQ*Q
      IF(PRODQ.GE. 1.D-35) GO TO 630
      PRODQ=PRODQ*1.D+35
      JEXPQ=JEXPQ+1
      GO TO 630
  401 M=M+1
      KRI=KILLS(M)
      GO TO 400
  206 P=CK
      IF(PSIARG.LT. 0.) P=.9999999
      GO TO 202
 3262 IF(IFIRST.EQ.1) GO TO 630
      FACT=Q
      IF(JRIC1.EQ.1) FACT=P
 3263 PROD=PROD*FACT
      IF(PROD.GT. 1.D-35) GO TO 630
      IEXP=IEXP+1
      PROD=PROD*1.D+35
  630 PSI=P-CK
      TBSQ=TB*TB
      TC=Q/P
      TD=TC*PSI
      IF(JRIQ1.NE.0) GO TO 3020
      TA=P
      TE=-PSI
      F121=F121-PSI
      GO TO 3023
 3020 TQ=RIQ1/P
      TA=P-TQ*CK
      T1=TD*TQ
      IF(IC.NE.3) GO TO 3021
      U3=U3+TQ
      F13=F13+T1*TB
      F23=F23+T1
      F33=F33+TQ*(TC-1.)
 3021 TE=(TQ-1.)*PSI
      F121=F121+TE
 3023 T1=TA*TD
      U2=U2+TE
      F11=F11+TBSQ*T1
      F22=F22+T1
      F122=F12_+TB*T1
      U1=U1+TE*TB
  400 CONTINUE
      T4=NPP
      JLI=1
```

```
      T1=1.7C2*QCI
      T2=T1*T1
      DT1=-T1*AK
      U2=U2*DT1
      F22=F22*(DT1*DT1)
      IF(IC-2) 41C0,410,4101
 4101 U3=(U3-T4)*QCI
      TT1=T1*QCI
      F13 =F13*TT1
      F23=-F23*(AK*TT1)
      F33=(F33+T4)*(QCI*QCI)
  410 U1=U1*T1
      F11=F11*T2
      F121=T1*F121-F122*(T2*AK)
 4100 IF(ISTAGE.GT. ILAST) GO TO 1604
C     WRITE MATRIX OF SECOND DERIVATIVES
      IF(IC-2) 1601,1602,1603
 1601 WRITE(6,9109) F22,U2
 9109 FORMAT(' ',D20.6,2X,D20.6)
      GO TO 1604
 1602 WRITE(6,169C) F11,F121,U1,F22,U2
 1690 FORMAT(' ',2D20.6,2X,D20.6/21X,D20.6,2X,D20.6)
      GO TO 1604
 1603 WRITE(6,1691) F11,F121,F13,U1,F22,F23,U2,F33,U3
 1691 FORMAT(' ',3D20.6,2X,D20.6/21X,2D20.6,2X,D20.6/41X,D20.6,2X,D20.6)
 1604 INDB=0
      INDA=0
      IND=0
      SAVE7=U1
      SAVE8=U2
      SAVE9=U3
      S3=0.
      S1=0.
 6010 IF(IC.EQ.1) GO TO 6020
      A11=F11
      DT1=F121/F11
      A22=F22-DT1*F121
      SS2=U2-DT1*U1
      IF(IC.EQ.2) GO TO 6004
      A23=F23-DT1*F13
      DT1=F13/F11
      A33=F33-DT1*F13
      SS3=U3-DT1*U1
      DT1=A23/A22
      A33=A33-A23*DT1
      SS3=SS3-SS2*DT1
      IF(A33-0.DC) 6003,440,6004
 6003 A33=-A33
      IND=-3
 6004 IF(A11-0.DC) 6005,44C,6006
```

```
 6005 A11=-A11
      IND=-1
 6006 IF(A22-0.D0) 6007,4420,6008
 6007 A22=-A22
      IND=-2
 6008 IF(IC-2) 6020,4420,6012
 6012 SSC=SS3/A33
      SSB=(SS2-SSC*A23)/A22
      S1=(U1-SSB*F121-SSC*F13)/A11
      S2=SSB
      S3 = SSC
      GO TO 1605
 4420 SSB=SS2/A22
      S1=(U1-SSB*F121)/A11
      S2=SSB
      GO TO 1606
 6020 S2=U2/F22
      IF(F22.GT.0.DC) GO TO 1607
      IND=-1
      S2=-S2
      GO TO 1607
 1605 CK=CK+S3
 1606 AK=AK+S1
 1607 BK=BK+S2
      SA1=AK
      SA2=S1
      SA3=BK
      SA4=S2
      SA5=CK
      SA6=S3
 1460 ICA=IC
      IE=IC
 1630 INDIC=0
      IF(ICA.LT.3) GO TO 1480
      SAAB=ABS(S3)
C      CHECK MAXIMUM STEP SIZE FOR C
      IF(SAAB.LE. CLAMDA) GO TO 1100
      ADJ(6)=STAR
      XMULT=CLAMDA/SAAB
      T3=CLAMDA
      IF(S3.LT. 0.) T3=-T3
      S3=T3
      NOAD=1
      CK=SAVEC+S3
      S2=S2*XMULT
      BK=SAVEB+S2
      IF(ICA.EQ.4) GO TO 1100
      S1=S1*XMULT
      AK=SAVEA+S1
 1100 CP=CK
```

```
      IF(CK.LT.0.) CK=C.
C     SEE IF CK IS WITHIN THE RANGE OF 0 TO .5
      IF(CK.GT. .5) CK=.5
C     IS CK ALLOWED TO VARY OR IS IT TO BE RESTRICTED
      T3=1.D0/A33
      SCC=SQRT(T3)
      IF(IND.LT.0) GO TO 1618
      IF(ID.LT. 0) GO TO 1618
 1108 CT=ABS(CK-CBAR)/SCC
      IF(CT.LT. .9) GO TO 1611
C     AT 1611 SET CK TO CBAR
      IF(CT .LT. 1.1) GO TO 1612
C      AT 1612 SEE IF CK =CBAR, IF SO SET CK TO CBAR
 1621 IF(CK .LE. CLO) GO TO 1613
C     AT 1613 SEE IF CK DIFFERS FROM CLO BY MORE THAN 2 STD. DEV.
      IF(CK .LT. CHI) GO TO 1618
C     AT 1618 ACCEPT CK
C     SEE IF CHANGE IN C IS NEGATIVE, IF IT IS SEE IF IT DIFFERS FROM
C        ORIGINAL VALUE
      IF(S3.LT.0.) GO TO 1618
C     SEE IF DIFFERENCE BETWEEN CURRENT C AND PREVIOUS C IS MORE THAN
C        2 STANDARD DEVIATIONS
      CT=(CK-SAVEC)/SCC
      IF(CT .GT. 2.) GO TO 1618
C     IF SO ACCEPT CK, IF NOT SET C TO PREVIOUS VALUE AND RESOLVE FOR
C        A AND B OR B
 1620 CK=SAVEC
      S3=0.
      ADJ(5)=STAR
C      C IS HELD FIXED, RESOLVE FOR A AND B
      IF(IND.LT.0) GO TO 1622
      WRITE(6,6301) IK,ITER,SA1,SA2,SA3,SA4,SA5,ADJ(5),SA6,ADJ6,SCC,
     1HOLL(3),CK
 6301 FORMAT(' ',I4,I3,3OX,F9.4,1X,F10.6,1X,F9.4,1X,F10.6,1X,F8.4,A1,
     1F10.6,A1,F10.6,1X,A1,1X,F6.4)
      GO TO 1623
 1622 WRITE(6,6302) IK,ITER,SA1,SA2,SA3,SA4,SA5,ADJ5,SA6,ADJ6,SCC,
     1HOLL(3),CK
 6302 FORMAT(' ',I4,I3,3OX,F9.4,1X,F10.6,1X,F9.4,1X,F10.6,1X,F8.4,A1,
     1F10.6,A1,'(',F9.6,')' ,A1,1X,F6.4,1X,'****')
 1623 NUAD=C
      I2=2
      SA5=CK
      SA6=S3
      SAVE9=C.
      DO 1624 J=1,6
       ADJ(J)=BLANK
 1624 CONTINUE
      SCC=0.
      IF(ICA.EQ.4) GO TO 1615
```

```
      SSB=SS2/A22
      S1=(U1-SSB*F121)/A11
      S2=SSB
      BK=SAVEB+S2
      AK=SAVEA+S1
      SA1=AK
      SA2=S1
      SA3=BK
      SA4=S2
      ICA=2
      GO TO 1480
 1615 S2=U2/F22
      BK=SAVEB+S2
      SA3=BK
      SA4=S2
      IE=1
      GO TO 1485
 1611 CK=CBAR
 1618 IF(ABS(CK-SAVEC) .LT. .000001) GO TO 1620
C     IF EQUALS PREVIOUS VALUE RESOLVE FOR A AND B
      IF(ABS(CK-CLDC) .GT. .000005) GO TO 1616
C      CK EQUALS PREVIOUS C, HOLD C FIXED AND RESTART ITERATIONS
C     CRITICAL RATIO IS LESS THAN 1.1, IF PREVIOUS C=CBAR, SET
C        CURRENT C TO CBAR
      S3=CK-SAVEC
      IF(CK.EQ.CP) GO TO 205
      NOAD=1
      ADJ(5)=STAR
  205 IF(IND.LT.0) GO TO 2633
      IF(NOAD.EQ.0) GO TO 6011
      WRITE(6,9485) IK,ITER,SA1,ADJ1,SA2,ADJ2,SA3,ADJ3,SA4,ADJ4,SA5,
     1ADJ5,SA6,ADJ6,BLANK
 6011 WRITE(6,6300)IK,ITER,IE,SAVE7,SAVE8,SAVE9,AK,S1,BK,S2     ,
     1CK,S3,SCC,BLANK
 2634 CK=OLDC
      AK=OLDA
      BK=OLDB
      SL=BLANK
      WRITE(6,4004) IK,AK,BK,CK
 4004 FORMAT(' *',I3,33X,F9.4,13X,F8.4,13X,F7.4)
      IC=2
      DO 4001 JJ=3,11
      RR(JJ)=0.
 4001 CONTINUE
      ITER=0
      LBEG=LBEG-NLP
      GO TO 301
 2633 IF(NOAD.EQ.0) GO TO 6013
      WRITE(6,9485) IK,ITER,SA1,ADJ1,SA2,ADJ2,SA3,ADJ3,SA4,ADJ4,SA5,
     1ADJ5,SA6,ADJ6,STARR
```

```
 6013 WRITE(6,6300)IK,ITER,IC,SAVE7,SAVE8,SAVE9,AK,S1,BK,S2     ,
     1CK,S3,SCC,STARR
      GO TO 2634
  440 WRITE(6,2440) IK
 2440 FORMAT(' **** DETERMINANT OF THE MATRIX OF SECOND DERIVATIVES IS Z
     1ERO FOR ITEM ',I6,'  PARAMETERS HELD CONSTANT FOR THIS STAGE')
      AK=SAVEA
      BK=SAVEB
      CK=SAVEC
      GO TO 490                                                    LOGI 656
 1612 IF (SAVEC .EQ. CBAR) GO TO 1611
C     OTHERWISE CHECK FOR GOING OUTSIDE MAX OR MIN
      GO TO 1621    -
C     SEE IF CK DIFFERS FROM CLO BY MORE THAN 2 STD. DEV.
 1613 CT=(CLC-CK)/SCC
      IF(CT.GT. 2.) GO TO 1618
C     SET CK=CLO
      CK=CLO
      GO TO 1618
C     SEE IF CK DIFFERS FROM CP
 1616 IF(CK.EQ. CP) GO TO 1480
 1104 DELX=CK-SAVEC
      ADJ(5)=STAR
      NOAD=1
      R2=DELX/S3
      XMULT=ABS(R2)
      S3=DELX
      IF(R2.LT.0 .OR. XMULT .GT. 1.) GO TO 1617
      S2=S2*XMULT
      BK=SAVEB+S2
      IF(ICA.EQ.4) GO TO 1480
      S1=S1*XMULT
      AK=SAVEA+S1
C     RESTRICTIONS ON A AND DELTA A
 1617 INDIC=1
 1480 IF(ICA.EQ.1 .OR. ICA.EQ.4) GO TO 1475
C     CHECK MAXIMUM STEP SIZE FOR A
      ALAMDA=ABS(.1*SAVEA)+.2
      IF(ABS(S1) .LT. ALAMDA) GO TO 1470
      ADJ(2)=STAR
      XMULT=ABS(ALAMDA/S1)
      NOAD=1
      IF(S1.LT.0.) ALAMDA=-ALAMDA
      S1=ALAMDA
      AK=SAVEA+S1
      S2=S2*XMULT
      BK=SAVEB+S2
      IF(ICA.LT.3) GO TO 1470
      IF(INDIC.EQ.1) GO TO 1470
      S3=S3*XMULT
```

```
      CK=SAVEC+S3
 1470 IF(AK.LT. AHI) GO TO 4480
C      AK IS LESS THAN MAXIMUM
      IF(SAVEA.LT.AHI) GO TO 4480
C      A IS BEING HELD FIXED, RESOLVE FOR B AND C, OR B ONLY
      SET=AHI
  506 ADJ1=STAR
      IF(IND.LT.0) GO TO 1626
      WRITE(6,6304)IK,ITER,SA1,ADJ1,SA2,ADJ2,SA3,ADJ3,SA4,ADJ4,SA5,ADJ5,
     1SA6,ADJ6,SCC,HOLL(1),SET
      GO TO 1627
 1626 WRITE(6,6305)IK,ITER,SA1,ADJ1,SA2,ADJ2,SA3,ADJ3,SA4,ADJ4,SA5,ADJ5,
     1SA6,ADJ6,SCC,HOLL(1),SET
 6304 FORMAT(' ',I4,I3,30X,F9.4,A1,F10.6,A1,F9.4,A1,F10.6,A1,F8.4,A1,
     1F10.6,A1,F10.6,1X,A1, 1X,F6.4)
 6305 FORMAT(' ',I4,I3,30X,F9.4,A1,F10.6,A1,F9.4,A1,F10.6,A1,F8.4,A1,
     1F10.6,A1,'(',F9.6,')' ,A1,1X,F6.4,1X,'****')
 1627 DO 1511 L=1,6
 1511 ADJ(L)=BLANK
      AK=SET
      S1=0.
      SA1=AK
      SA2=S1
      NOAD=0
      IF(ICA.EQ.2) GO TO 1631
      DT1=F23/F22
      A33=F33-DT1*F23
C     CHECK FOR NON POSITIVE DEFINITE MATRIX
      IF(A33-0.D0) 1633,440,1634
 1633 IND=-3
      A33=-A33
 1634 IF(F22-0.) 1635,440,1636
 1635 IND=-2
      F22=-F22
 1636 SS3=(U3-DT1*U2)/A33
      S2=(U2-SS3*F23)/F22
      S3=SS3
      ICA=4
      BK=SAVEB+S2
       CK=SAVEC+S3
      SA3=BK
      SA4=S2
      SA5=CK
      SA6=S3
      IE=2
      GO TO 1630
C      SOLVE FOR B ONLY
 1631 S2=U2/F22
      IF(F22.GT.0.D0) GO TO 1632
      S2=-S2
```

```
      IND=-2
 1632 BK=SAVEB+S2
      SA3=BK
      SA4=S2
      IE=1
      GO TO 1475
C     CHECK DELTA B AT 1475
C      CHECK BOUNDS ON A
 4480 IF(AK .GE. ALO) GO TO 501
      IF(SAVEA.NE. ALO) GO TO 507
      SET=ALO
      GO TO 506
  507 AK=ALO
      GO TO 503
  501 IF(AK.LE. AHI) GO TO 1475
      AK=AHI
  503 DELX=AK-SAVEA
      XMULT=ABS(DELX/S1)
      NOAD=1
      S1=DELX
      ADJ(1)=STAR
      S2=S2*XMULT
      BK=SAVEB+S2
      IF(IC.LT.3) GO TO 1475
      IF(INDIC.EQ.1) GO TO 1475
      S3=S3*XMULT
      CK=SAVEC+S3
C      CHECK MAXIMUM STEP SIZE ON B
 1475 BLAMDA=ABS(.2*SAVEB)+.4
      IF(ABS(S2).LT. BLAMDA) GO TO 1485
      NOAD=1
      ADJ(4)=STAR
      XMULT=ABS(BLAMDA/S2)
        IF(S2.LT.0.) BLAMDA=-BLAMDA
      S2=BLAMDA
      BK=SAVEB+S2
      IF(ICA.EQ.1) GO TO 1485
      IF(ICA.EQ.4) GO TO 1640
      S1=S1*XMULT
      AK=SAVEA+S1
      IF(ICA.LT.3) GO TO 1485
 1640 IF(INDIC.EQ.1) GO TO 1485
      S3=S3*XMULT
      CK=SAVEC+S3
 1485 IF(NOAD.EQ. 0) GO TO 480
      SL=BLANK
      IF(IND.LT.0) SL=STARR
      WRITE(6,9485) IK,ITER,SA1,ADJ1,SA2,ADJ2,SA3,ADJ3,SA4,ADJ4,SA5,
     1ADJ5,SA6,ADJ6,SL
 9485 FORMAT(' ',I4,I3,30X,F9.4,A1,F10.6,A1,F9.4,A1,F10.6,A1,F8.4,
```

```
     1A1,F10.6,A1,20X,A4)
  480 IF(IC-2) 3017,3018,3019
 3019 T1=ABS(SAVE9)
      IF(T1.GT. RDC) RDC=T1
      SC=SC+S3
      T1=ABS(S3)
      IF(T1.GT. ABS(RC)) RC=S3
 3016 T1=ABS(S1/AK)
      IF(T1.GT. ABS(RA)) RA=S1/AK
      T1=ABS(SAVE7)
      IF(T1.GT. RDA) RDA=T1
      SA=SA+S1/AK
 3017 T1=ABS(S2)
      IF(T1.GT. ABS(RB)) RB=S2
      T1=ABS(SAVE8)
      IF(T1.GT. RDB) RDB=T1
      SB=SB+S2
      IF(IND.LT.0) GO TO 6040
      IF(ID.LT.0 .AND. ID .GT. (-10)) ID=10-ID
      GO TO 6041
 6040 IF(ID.GT.0) ID=-ID
 6041 IF(IND.LT.0) GO TO 2621
      WRITE(6,6300) IK,ITER,IE,SAVE7,SAVE8,SAVE9,AK,S1,
     1BK,S2,CK,S3,SCC,BLANK
 6300 FORMAT(' ',I4,2I3,3F9.4,F9.4,1X,F10.6,2X,F8.4,1X,F10.6,2X,F7.4,
     111X,F10.6,1X,F10.6,10X,A4)
 2623 IF(ITER.LT. MAXIT) GO TO 481
      GO TO 490
 2621 IE=-IE
      WRITE(6,2622) IK,ITER,IE ,SAVE7,SAVE8,SAVE9,AK,S1,
     1BK,S2,CK,S3,SCC,STARR
 2622 FORMAT(' ',I4,2I3,3F9.4,F9.4,1X,F10.6,2X,F8.4,1X,F10.6,2X,F7.4,
     1 1X,F10.6,1X,'(',F9.6,')',9X,A4)
      GO TO 2623
  300 LBEG=LBEG-NLP
      GO TO 301
 5001 MI=MI+1
      KRE=KITE(MI)
      LBEG=LBEG+NLP
      GO TO 500
  481 IF(ABS(S1/AK) .GE. TOLA) GO TO 300
      IF(ABS(S3).GE. CLAMDA) GO TO 300
      IF(ABS(S3).LT. .01) GO TO 490
      IF((ITER-2*(ITER/2)) .NE. 0) GO TO 300
  490 IF(ITER.GT. ITMAX) ITMAX=ITER
  532 IF(AK.LT. AHI) GO TO 131
      XCC=XINV
      IF(NIT.EQ.0) XCC=1.
      AK=AHI*XCC
  131 A(K)=AK
```

```
      B(K)=BK
      C(K)=CK
      VA=(A(K)-UA(K))/A(K)
      VB=B(K)-UB(K)
      VC=C(K)-UC(K)
      DO 4002 JJ=3,11
      T1=ABS(PR(JJ))
      IF(T1.LE. ABS(V(JJ)))GO TO 4002
      V(JJ)=RR(JJ)
      JS(JJ)=IK
 4002 CONTINUE
      KDD(K)=ID
      NOM(K)=NO
      NPPI(K)=NPP
      DA(K)=ABS(SA)
      DB(K)=ABS(SB)
      DC(K)=ABS(SC)
      IF(INDIC.EQ.1) INDCC=1
  500 CONTINUE                                                          LOGI 845
      WRITE(6,9021)ISTAGE
 9021 FORMAT('1 STAGE ',I4/'0ITEM(K)   A(R)      A(R)-A(R-1)',6X,'B(R)
     1  B(R)-B(R-1)',8X,'C(R)    C(R)-C(R-1) ',                        'PA
     1RAM CODE NO. PEOPLE NO. OMITS      ')
 9022 FORMAT(' ',I6,2F12.6,2X,2F12.6,2X,2F12.4,6X,I4,6X,I4,6X,I4)
      MI=1
      KRE=KITE(1)
      DO 292 K=1,LITTLN
      IF(ITEMNC(K).EQ.KRE ) GO TO  2920
      VA=(A(K)-UA(K))/A(K)
      VB=B(K)-UB(K)
      VC=C(K)-UC(K)
      UA(K)=A(K)
      UB(K)=B(K)
      UC(K)=C(K)
      WRITE(6,9022) ITEMNO(K),A(K),VA,B(K),VB,C(K),VC,KDD(K), NPPI(K),
     1NOM(K)
      GO TO 292
 2920 MI=MI+1
      KRE=KITE(MI)
  292 CONTINUE
      LOG=2
      IF(IFIRST.NE.1) GO TO 3207
 3206 PRODS=PROD
      JS(15)=NESTH
      IF(JSUMC.GT. 248) GO TO 2702
      DO 2701 K=1,13
      JSUMC=JSUMC+1
      VV(JSUMC)=V(K)
      JSA(JSUMC)=JS(K)
 2701 CONTINUE
```

```
      JSA(JSUMC+1)=JS(14)
      JSUMC=JSUMC+2
      JSA(JSUMC)=JS(15)
 2702 WRITE(6,2703)
 2703 FORMAT('1MAXIMUM DERIVATIVE',26X,'MAXIMUM CHANGE',24X,'MAXIMUM TOT
     1AL CHANGE')
      WRITE(6,2500)(HCLL(I),V(I),JS(I),I=3,5)
      WRITE(6,2500)(HCLL(2),V(I),JS(I),I=6,8)
      WRITE(6,2500)(HCLL(3),V(I),JS(I),I=9,11)
       TOLA=.3*ABS(V(4))
      IF(TOLA.LT. .C01) TOLA=.C01
      DAA=.2*ABS(V(5))
      DBB=.2*ABS(V(8))
      DCC=.2*ABS(V(11))
 2500 FORMAT(6X,A1,1X,F13.8,' FOR ITEM ',I3,8X,2(6X,A2,F13.8,' FOR ITEM
     1',I3,8X))
      WRITE(6,2501) TOL,JDRTH,THCH,JCHTH
 2501 FORMAT('  THETA ',F13.8,' FOR EXAMINEE ',I4,4X,' THETA ', F13.8,
     1' FOR EXAMINEE ',I4)
      WRITE(6,2704)ITMAX,ITMAXT
 2704 FORMAT('0MAXIMUM ITERATIONS -- FOR ITEMS ',I2,', -- FOR THETAS ',
     1I2)
      IF(IFIRST.EQ.1) GO TO 241
      J=ISTAGE
      IF(NIT.EQ.0) J=J-1
      WRITE(6,2003)J,PROD,IEXP
 2003 FORMAT('0*****  CRITERION FUNCTION AFTER THETA ESTIMATION IN STAGE
     1 ',I4,' *****  L=',F15.8,'*10**(',I10,')')
  241 CALL RESCLK(2,TIME)
      JFIRST=IFIRST
      SECS=CLOCK(ELA)
      SUBTOT=(STIME-SECS)*.1E-5
      IF(IFIRST.EQ. 0 .AND. NIT.EQ.1) TOLTOT=SUBTOT
      IF(NIT.EQ.1) IFIRST=1
      IF(ABS(V(5)).LE. ACRIT) IFIRST=2
      CALL CLOSE(11)
      NIT=1
  660 WRITE (6,9020) ISTAGE                                          LOGI 854
 9020 FORMAT (1H0,12HEND OF STAGE,I3)
      CALL CLOSE(12)
      IF (ISTAGE.EQ.MAXST) GO TO 900                                 LOGI 860
      IF(IQUIT.LE.0 .AND. INDCC.EQ.0) GO TO 921
  800 WRITE (6,2910) SUBTOT                                          LOGI 867
 2910 FORMAT (1H0,17HTIME FOR STAGE = ,F7.2,8H SECONDS  /'1')
      TIMTOT=TIMTOT+SUBTOT
      IF((TIMTOT+TOLTOT).LT. TIMAX) GO TO 805
      DO 801 L=1,5                                                   LOGI 870
  801 WHY(L)=TIMEC(L)                                                LOGI 871
      GO TO 900                                                      LOGI 872
  805 IF(ICKA.NE.0) GO TO 502
```

```
C     REMOVE ITEMS IF NECESSARY
      MI=1
      KRI=KITE(1)
      L=0
      DO 810 K=1,LITTLN
      IK=ITEMNO(K)
      IF(IK.EQ.KRI) GO TO 815
      IF(K.EQ.KEEP) GO TO 121
      IF(ABS(A(K)) .LE. ALIM) GO TO 810
  121 L=L+1
      NOM(L)=IK
      WRITE(6,2185) IK
      IF(L .LE. MAXOUT) GO TO 810
      WRITE(6,9815)
 2185 FORMAT (1H0,5H*****,5H ITEM,I4,16H WILL BE REMOVED  )          LOGI 883
 9815 FORMAT (1H0,38HMAX. NO. OF ITEMS HAVE BEEN REMOVED *  )       LOGI 887
      GO TO 910
  815 L=L+1
      NOM(L)=KRI
      MI=MI+1
      KRI=KITE(MI)
  810 CONTINUE
      IF(L.EQ.NKITE) GO TO 502
      IFIRST=0
      NKITE=L
      DO 811 J=1,L
      KITE(J)=NOM(J)
  811 CONTINUE
  502 CALL REOPEN(11)
      CALL REOPEN(12)
      GO TO 50
  921 MAXST=MAXST+ISTAGE
      IEND=1
 6000 CALL PRODUC(PROD,IEXP)
      NEST=N-NKILL
 2896 WRITE(6,2898)
 2898 FORMAT('1FINAL THETAS',20X,'THETAS FOR REMOVED EXAMINEES, THETAS .
     1LE. -10, AND THETAS .GE. 100'/43X,'WILL BE PRINTED AS ASTERISKS')
      DO 267 I=1,N,25
      J=I+24
      IF(J.GT.N) J=N
       WRITE(6,2030) I,(THETA(K),K=I,J)
  267 CONTINUE
      MLEST=LITTLN-NKITE
      WRITE(7,534)TITLE,LITTLN,MLEST,N,NEST,DATER,ISTAGE
  534 FORMAT(15A4,2I3,2I6/5A4,4I5)
      WRITE(7,2844) PROD,IEXP,OLDSUM,KEXP,TOLA
 2844 FORMAT(A8,4A4)
      K=0
      DO 315 I=1,N,19
```

```
      K=K+1
      DO 6015 L=1,19
      J=I+L-1
      THETT(L)=THETA(J)
      IF(J.EQ.N) GO TO 6016
 6015 CONTINUE
      WRITE(7,311)K,THETT
  315 CONTINUE
      GO TO 6017
 6016 WRITE(7,311)K,(THETT(J),J=1,L)
 6017 L=1000
      DO 122 K=1,LITTLN,6
      J=K+5
      IF(J.GT. LITTLN) J=LITTLN
      L=L+1
      WRITE(7,123) L,(A(LL),B(LL),C(LL),LL=K,J)
  123 FORMAT(I4,18A4)
  122 CONTINUE
      WRITE(7,1001)(KDC(K),K=1,LITTLN)
      WRITE(7,1005)(ITEMNO(K),K=1,LITTLN)
      WRITE(7,1001) NKILL,(KILLS(M),M=1,NKILL)
      WRITE(7,1001) NKITE,(KITE(M),M=1,NKITE)
      WRITE(6,2705)
 2705 FORMAT('1',20X,'****       SUMMARY       ****')
      JSUMC=1
      JSTAGE=JSTAGE+1
      DO 2706 J=JSTAGE,ISTAGE
      LEND=JSUMC+1
      WRITE(6,2707)J,(VV(L),JSA(L),L=JSUMC,LEND)
 2707 FORMAT(///' STAGE ',I4/'  CRITERION  FUNCTIONS -- AFTER ITEM ESTIM
     1ATION IN PREVIOUS STAGE = ',F16.8,'*10**(',I10,')'/26X,'AFTER THET
     2A ESTIMATION IN THIS STAGE',4X,'= ',F16.8,'*10**(',I10,')')
      WRITE(6,2708)
 2708 FORMAT('0 MAXIMUM DERIVATIVE',26X,'MAXIMUM CHANGE',24X,'MAXIMUM TO
     1TAL CHANGE')
      DO 533 K=1,3
      JSUMC=LEND+1
      LEND=JSUMC+2
      WRITE(6,2500)(HOLL(K),VV(L),JSA(L),L=JSUMC,LEND)
  533 CONTINUE
      JSUMC=LEND+1
      LEND=JSUMC+1
      WRITE(6,2501)(VV(L),JSA(L),L=JSUMC,LEND)
      JSUMC=LEND+1
      WRITE(6,1300)JSA(JSUMC),JSA(JSUMC+1)
 1300 FORMAT('0',I6,' THETAS WERE ESTIMATED',10X,'NEW PARAMETERS WERE ES
     1TIMATED FOR ',I4,' ITEMS')
      JSUMC=JSUMC+2
      IF(JSUMC.GT. 246) GO TO 2709
 2706 CONTINUE
```

```
2709 IF(IEND-1) 800,8CC,990
 904 IF (IHAVE.EQ.1) GO TO 800                                          LOGI 943
 611 IHAVE=1                                                            LOGI 944
     MAXST=ISTAGE+2
     WRITE (6,2901)                                                     LOGI 947
2901 FORMAT('C L HAS DECREASED-- 2 MORE STAGES WILL BE DONE')
     DO 922 L=1,5
 922 WHY(L)=CONVR(L)                                                    LOGI 951
     GO TO 800                                                          LOGI 952
 900 WRITE (6,2010) ISTAGE,WHY                                          LOGI 964
2010 FORMAT (1HO,I3,17H STAGES COMPLETED,5A4)                           LOGI 965
 910 IEND=2
     GO TO 6000
 990 CALL RESCLK (3,TIME)                                               LOGI 975
     CALL EXIT
7653 FORMAT('  NORMALIZED A AND B ')
7654 FORMAT(' ',I4,2F12.6)
 311 FORMAT(I4,19A4)
9910 FORMAT(5E15.8)
9920 FORMAT(8F10.7)
1001 FORMAT(16I5)
1005 FORMAT(26I3,2X)
2030 FORMAT(' ',I4,2X,25F5.2)
1002 FORMAT(10A8)
     RETURN
     END                                                                LOGI 979
     SUBROUTINE PRODUC(PROD,IEXP)
     COMMON AHI,ALIM,ALO,ACRIT  ,CBAR,CGUESS,CHI,CHOICE,CLAMDA,CLO,
    1ICKA,ILAST,INDIC,ISTAGE,          JSTAGE,LITTLN,MAXOUT,MAXST,
    5MAXIT,IWILL,ITONE,NOUT,NRECPI,CLDSUM,KEXP,
    2N,NC,NIPREC,NIT,NKILL,NLB,NLP,NPPREC,    TIMAX,    TOL1,TOLA,
    3TOTAL,A(100),B(100),C(100),KILLS(100),KDO(100),ITEMNO(100),
    4IY,NKITE,KITE(100),TITLE(15),DATER(5),THETA(3000)
     REAL*8 TITLE
     REAL*8 PRODA,PROD
2850 PROD=1.00
     IEXP=0
     DO 2842 J=1,N
     IF(THETA(J).EQ. 0.) GO TO 2842
     PROD=PROD*THETA(J)
     PRODA=DABS(PROD)
     IF(PRODA.GE. 1.D-35) GO TO 2843
     PROD=PROD*1.D+35
     IEXP=IEXP-1
     GO TO 2842
2843 IF(PRODA.LE. 1.D+35) GO TO 2842
     PROD=PROD*1.D-35
     IEXP=IEXP+1
2842 CONTINUE
     IEXP=IEXP*35
```

```
      WRITE(6,2845) PROD,IEXP
 2845 FORMAT('0CHECK PRODUCT OF THETAS IS ',D24.16,'*10**(',I10,')')
       RETURN
      END
      SUBROUTINE MODIFY(ITEST,T1,T2,T3,J1,J2,MI,MLITN)
C      MODIFY ITEM PARAMETERS READ IN
      COMMON AHI,ALIM,ALO,ACRIT  ,CBAR,CGUESS,CHI,CHOICE,CLAMDA,CLO,
     1ICKA,ILAST,INDIC,ISTAGE,          JSTAGE,LITTLN,MAXOUT,MAXST,
     5MAXIT,IWILL,ITONE,NOUT,NRECPI,OLDSUM,KEXP,
     2N,NC,NIPREC,NIT,NKILL,NLB,NLP,NPPREC,    TIMAX,    TOL1,TOLA,
     3TOTAL,A(100),B(100),C(100),KILLS(100),KDO(100),ITEMNO(100),
     4IY,NKITE,KITE(100),TITLE(15),DATER(5),THETA(3000)
      DIMENSION T1(1),T2(1),T3(1),J1(1),J2(1),MI(1)
      DIMENSION ITEST(1)
      IF(MI(2) .EQ. 0) GO TO 711
      READ (5,1001) NOUT
      READ(5,7004)(J1(J),T1(J),T2(J),T3(J),J2(J),J=1,NOUT)
 7004 FORMAT(I5,3E15.8,I5)
 7003 FORMAT(I5/I5,E15.8,I5,E15.8,I5,E15.8,I5,E15.8)
C      ADD ITEMS--READ IN ITEM NUMBER, A , B , C ,
      WRITE(6,6001)
      WRITE(6,6000)(J1(J),T1(J),T2(J),T3(J),J2(J),J=1,NOUT)
 6000 FORMAT(' ',I5,3E16.8,6X,I5)
 6001 FORMAT('0THE FOLLOWING ITEMS AND THEIR PARAMETERS WILL BE ADDED'/
     1' ITEMNO',7X,'A',15X,'B',15X,'C',8X,'PARAM CODE')
      M=0
      NEQ=0
      DO 62 ME=1,NKITE
      KRE=KITE(ME)
      DO 60 J=1,NOUT
      IF(J1(J)-KRE) 60,61,63
   60 CONTINUE
   63 M=M+1
      KITE(M)=KRE
      GO TO 62
   61 NEQ=NEQ+1
   62 CONTINUE
      JK=NOUT
      JM=MLITN
      MLITN=MLITN+NOUT-NEQ
      DO 3111 J=1,MLITN
      JL=MLITN-J+1
      IF(J1(JK).GE.ITEMNO(JM)) GO TO 3112
      ITEMNO(JL)=ITEMNO(JM)
      A(JL)=A(JM)
      B(JL)=B(JM)
      C(JL)=C(JM)
      ITEST(JL)=ITEST(JM)
      JM=JM-1
      GO TO 3111
```

```
 3112 ITEMNO(JL)=J1(JK)
      A(JL)=T1(JK)
      B(JL)=T2(JK)
      C(JL)=T3(JK)
      ITEST(JL)=J2(JK)
      JK=JK-1
      IF(JK.EQ.0) GO TO 711
 3111 CONTINUE
      DO 3114 J=1,JK
      ITEMNO(J)=J1(J)
      A(J)=T1(J)
      B(J)=T2(J)
      C(J)=T3(J)
      ITEST(J)=J2(J)
 3114 CONTINUE
  711 IF(MI(3).EQ.0) GO TO 714
C     REMOVE ITEMS,  READ IN NUMBER OF ITEMS TO BE REMOVED
      READ(5,1001)NOUT,(J1(I),I=1,NOUT)
      WRITE(6,6002) (J1(J),J=1,NOUT)
 6002 FORMAT('0 THE FOLLOWING ITEMS WILL BE REMOVED'/(2415))
      JI=NOUT
      JRI=J1(JI)
      JR=NKITE
      NKITE=NKITE+NOUT
      JE=NKITE
   53 KRI=KITE(JR)
   51 IF(KRI.GT.JRI) GO TO 52
C     ADD
      KITE(JE)=JRI
      JE=JE-1
      JI=JI-1
      IF(JI.EQ.0) GO TO 714
      JRI=J1(JI)
      GO TO 51
   52 KITE(JE)=KRI
      JE=JE-1
      JR=JR-1
      IF(JR.NE.0) GO TO 53
      DO 54 J=1,JE
      KITE(JE)=J1(J)
   54 CONTINUE
  714 IF(MI(4).EQ.0) GO TO 719
C      REPLACE A, B, C FOR SPECIFIED ITEMS
      WRITE(6,6003)
 6003 FORMAT('0THE PARAMETERS FOR THE FOLLOWING ITEMS WILL BE CHANGED'/
     1' ITEMNO',7X,'A',15X,'B',15X,'C',8X,'PARAM CODE')
      READ(5,1001) NOUT
      DO 716 J=1,NOUT
      READ(5,7004)K,AA,BB,CC,IJK
      WRITE(6,6006) K,AA,BB,CC,IJK
```

```
      DO 717 L=1,MLITN
      IF(ITEMNO(L).EQ.K) GO TO 718
  717 CONTINUE
      GO TO 716
  718 A(L)=AA
      B(L)=BB
      C(L)=CC
      ITEST(L)=IJK
  716 CONTINUE
  719 IF(MI(5).EQ.0) GO TO 723
C      REPLACE A
      READ(5,7003) NOUT,(J1(I),T1(I),I=1,NCUT)
      WRITE(6,6004)
      WRITE(6,6005)(J1(I),T1(I),I=1,NOUT)
 6004 FORMAT('0 THE FOLLOWING A''S WILL BE CHANGED')
 6005 FORMAT(' ',I5,F12.8,I5,F12.8,I5,F12.8,I5,F12.8,I5,F12.8)
      DO 720 I=1,NOUT
      IJ=J1(I)
      DO 721 J=1,MLITN
      IF(IJ.EQ. ITEMNO(J)) GO TO 722
  721 CONTINUE
      GO TO 720
  722 A(J)=T1(I)
  720 CONTINUE
  723 IF(MI(6).EQ. 0) GO TO 727
C     REPLACE B
      READ(5,7003) NOUT,(J1(I),T1(I),I=1,NCUT)
      WRITE(6,6006)
 6006 FORMAT('0 THE FOLLOWING B''S WILL BE CHANGED')
      WRITE(6,6005)(J1(I),T1(I),I=1,NOUT)
      DO 724 I=1,NOUT
      IJ=J1(I)
      DO 725 J=1,MLITN
      IF(IJ.EQ.ITEMNO(J)) GO TO 726
  725 CONTINUE
      GO TO 724
  726 B(J)=T1(I)
  724 CONTINUE
  727 IF(MI(7).EQ.0) GO TO 745
C     REPLACE C
      READ(5,7003) NOUT,(J1(I),T1(I),I=1,NCUT)
      WRITE(6,6007)
 6007 FORMAT('0THE FOLLOWING C''S WILL BE CHANGED')
      WRITE(6,6005)(J1(I),T1(I),I=1,NOUT)
      DO 728 I=1,NOUT
      IJ=J1(I)
      DO 729 J=1,MLITN
      IF(IJ.EQ.ITEMNO(J)) GO TO 730
  729 CONTINUE
      GO TO 728
```

```
  730 C(J)=T1(I)
  728 CONTINUE
  745 IF(MI(8).EQ.0) GO TO 749
C     CHANGE PARAM CODES FOR THE FOLLOWING C'S
      READ(5,7005) NOUT
 7005 FORMAT(16I5)
      READ(5,7005)(J1(K),J2(K),K=1,NOUT)
      WRITE(6,6010)(J1(K),J2(K),K=1,NOUT)
 6010 FORMAT('0THE PARAMETER CODES FOR THE FOLLOWING ITEMS WILL BE CHANG
     1ED'/' ITEMNO PARAM CODE'/(I5,5X,I5))
      DO 746 K=1,MLITN
      IT=ITEMNO(K)
      DO 747 L=1,NOUT
      JL=L
      IF(J1(L).EQ.IT) GO TO 748
  747 CONTINUE
      GO TO 746
  748 ITEST(K)=J2(JL)
  746 CONTINUE
  749 IF(MI(9).EQ.0) GO TO 752
C     CHANGE PARAM CODES FOR ALL ITEMS
      READ(5,7005)NOUT
      WRITE(6,6011) NOUT
 6011 FORMAT('0THE PARAMETER CODES FOR ALL ITEMS WILL BE CHANGED TO ',
     1I3)
      DO 750 K=1,MLITN
      ITEST(K)=NOUT
  750 CONTINUE
  752 RETURN
 1001 FORMAT(16I5)
      END
      SUBROUTINE RESCLK(I,TIME)
C     I=1--- INITIALIZE TIME
C     I=2--- ELAPSED TIME SINCE LAST CALL
C     I=3--- ELAPSED TIME SINCE INITIAL CALL
      IF(I-2)1,2,3
    1 T2=CLOCK(ELA)
      TIMBEG=T2
      GO TO 4
    2 T1=CLOCK(ELA)
      TIME=(T2-T1)*.1E-5
      WRITE(6,1000) TIME
 1000 FORMAT('0',F12.6,' SECONDS ELAPSED SINCE LAST CALL')
       T2=T1
    4 RETURN
    3 T1=CLOCK(ELA)
      TIME=(TIMBEG-T1)*.1E-5
      WRITE(6,1001) TIME
 1001 FORMAT('0',F12.6,' SECONDS ELAPSED SINCE INITIAL CALL')
      GO TO 4
```

```
      END
      FUNCTION PINV(PP)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PINV,PP
      DATA PI/3.14159265358979321
      P=PP
      J=0
      IF(P.LE.0.D0) GO TO 3
      IF(P.GE.1.D0 ) GO TO 4
      RTPI=1./DSQRT(2.D0*PI)
      SRT2=1./DSQRT(2.0D0)
      T=1.0D0-P
      T=DSQRT(DLOG(1.0D0/(T*T)))
      X1=T-((T*.010328D0+.802853D0)*T+2.515517D0)/(1.+((T*.00C13C8D0
     1+.189269D0)*T+1.432788D0)*T)
    1 J=J+1
      X2=X1-(.5D0*(DERF(X1*SRT2)+1.0D0) -P)/(RTPI*DEXP(X1*X1*(-.5D0)))
      IF(DABS(X2-X1).LT. .1D-8) GO TO 2
      IF (J.GT.20C) GO TO 2
      X1=X2
      IF (DABS(X1).GT. 6.0D0) GO TO 5
      GO TO 1
    2 PINV=X2
      RETURN
    3 X2=-6.D0
      GO TO 2
    4 X2=+6.0D0
      GO TO 2
    5 IF(P.LT..1D0) GO TO 3
      IF(P.GT. .9D0) GO TO 4
      END
MOV2    TITLE 'MOV2BI'
*       SUBROUTINE MOV2BI(N,C,M,D)
*       MOVES N-TH 2 BITS OF WORD C INTO M-TH  2 BITS OF WORD D
*       N=IDENTIFICATION OF 2 CONSECUTIVE BITS TO BE MOVED;0,2,...,30
*       C=WORD CONTAINING BITS TO BE MOVED
*       M=IDENTIFICATION OF DESTINATION OF BITS;0,2,...,30
*       D=WORD INTO WHICH THE BITS WILL BE MOVED
MOV2BI  START 0
        ENTRY  MOV2BI
        BC     15,12(15)
        DC     X'7'                 SET UP NAME
        DC     CL7'MOV2BI'
        STM    2,9,28(13)           STORE REGISTERS
        BALR   9,0                  SET UP 9 AS BASE REGISTER
        USING *,9
        L      R2,0(1)
        L      R1,0(R2)             R1 CONTAINS N
        STC    R1,ST1+3             STORE N IN SHIFT LEFT COMMAND
        L      R2,8(1)
```

```
         L      R1,0(R2)          R1 CONTAINS M
         A      R1,=F'2'          ADD 2 TO M
         STC    R1,ST2+3          STORE IN LEFT SHIFT DOUBLE
         STC    R1,ST3+3          STORE IN RIGHT SHIFT DOUBLE
         L      R2,4(1)           R2 CONTAINS ADDRESS OF C
         L      R4,12(1)          R4 CONTAINS ADDRESS OF D
         L      R5,C(R2)          R5=C
ST1      SLL    R5,0(0)           SHIFT LEFT N POSITIONS
         SRL    R5,30(0)          SHIFT RIGHT 30--SO THAT BITS WANTED
         L      R6,=F'0'          ARE IN POSITIONS 30 + 31
         L      R7,0(R4)          R7 CONTAINS  D
ST2      SLDL   R6,0(0)           SHIFT R6 LEFT M+2 POSITIONS
         N      R6,MASK           SET BITS 30+31 TO 0
         OR     R6,R5             ADD BITS FROM C
ST3      SRDL   R6,0(0)           SHIFT RIGHT M+2 POSITIONS
         ST     R7,0(R4)          STORE IN D
         LM     2,9,28(13)        RESTORE REGISTERS
         MVI    12(13),X'FF'      INDICATE CONTROL RETURNED
         BCR    15,14             RETURN
R1       EQU    8
R2       EQU    2
R3       EQU    3
R4       EQU    4
R5       EQU    5
R6       EQU    6
R7       EQU    7
         DS     0F
MASK     DC     X'FFFFFFFC'
         END
I2BI     TITLE 'I2BIT'
*  FUNCTION I2BIT(N,C)
*        RETRIEVE BIT N AND BIT N+1 FROM C
*        C=WORD CONTAINING DESIRED TWO BITS
*        N= IDENTIFICATION OF FIRST BIT OF TWO BITS
*             N=0,2,...,30 FORM LEFT TO RIGHT
I2BIT START 0
         ENTRY I2BIT
         BC     15,5+1+4(15)
         DC     X'5'
         DC     CL5'I2BIT'        SET UP NAME
         STM    2,3,28(13)        STORE TWO REGISTERS
         BALR   3,0
         USING *,3                SET UP REGISTER 3 AS BASE
         L      R2,0(1)           REGISTER
         L      R1,0(R2)          LOAD R1 WITH N
         STC    R1,STMT1+3        STORE IN SHIFT LEFT COMMAND
         L      R2,4(1)
         L      R1,0(R2)          LOAD R2 WITH C
STMT1    SLL    R1,0(0)           SHIFT  LEFT N POSITIONS
         SRL    R1,30(0)          SHIFT RIGHT 30 POSITIONS
```

```
         LM    2,3,28(13)          RESTORE REGISTERS
         MVI 12(13),X'FF'          INDICATE CONTROL RETURNED
         BCR   15,14               RETURN
R1       EQU   0
R2       EQU   2
         END
      FUNCTION CLOCK(ELA)
      INTEGER LAST/ZC4B20100/
      NOW=ITIME(DUM)
      ELA=(LAST-NOW)*26
      CLOCK=NOW*26
      RETURN
      END
ETSIO     CSECT                                                    00000010
         ENTRY OPEN,GET,PUT,CLOSE,REOPEN                           00000020
OPEN     SAVE  (14,12),,*                                          00000030
         BALR  12,0                                                00000040
         USING *,12                                                00000050
         ST    13,SAVE+4           SET UP TO SAVE AND RESTORE 13   C00CCC6C
          LR   HOLD,13                                            00000070
         LA    13,SAVE                                           00000080
         ST    13,8(0,HOLD)                                      00000090
         LA    TMP,4C(0,0)         SET TMP TO COUNT NUMBER OF BYTES 00000100
          LR   XR,1                XR POINTS TO ARGUMENT LIST     00000110
COUNT     CLI   0(XR),X'80'        LAST ARG?                      00000120
         LA    TMP,96(0,TMP)       ADD 96 BYTES FOR EACH DCB      00000130
         BE    ONWARD                                            00000140
         LA    XR,4(0,XR)          BUMP XR TO NEXT ARG.           00000150
          B    COUNT                                             00000160
ONWARD   LR    0,TMP               LOAD R0 WITH BYTES REQUIRED    00000170
         LR    TMP,1               POINT TMP TO ARG LIST          00000180
         S     TMP,FOUR
          USING DUM,11                                            00000190
         USING IHADCB,10                                          00000200
       GETMAIN R,LV=(0)                                           00000210
          LR   11,1                POINT R11 TO DUM               00000220
          LA   10,40(0,1)          POINT R10 TO DCB1              00000230
         LA    XR,0(0,0)           SET XR TO ZERO                 00000240
LOOP  LA       TMP,4(0,TMP)
         LA    9,TABL                                            00000260
         L     ARG,0(0,TMP)                                      00000270
          L    ARG,0(0,ARG)        GET ARGUMENT                   00000280
          LTR  ARG,ARG             CHECK FOR INPUT OR OUTPUT      00000290
         BP    POSET                                             00000300
         MVC   FLAG(2),OUT         SET FLAG FOR OUTPUT            00000310
         LA    FLAGR,15(0,0)                                     00000320
          LPR   ARG,ARG            MAKE ARGUMENT POSITIVE         00000330
         B     GOGO                                              00000340
POSET    MVC   FLAG(2),IN          SET FLAG FOR INPUT             00000350
         LA    FLAGR,0(0,0)                                      00000360
```

```
GOGO     S      ARG,TEN                                                  00000370
         SLA    ARG,2(0)                                                 00000380
         MVC    0(96,10),MODL       MOVE MODEL DCB                       00000390
STORE    ST     10,LST(ARG)         STORE DCB ADDRESS IN LIST(I)         00000400
         LA     2,CHECK
         AR     2,ARG               ADD DISP                             00000404
         OI     0(2),X'7F'          SET OPEN INDICATER                   00000406
          ST    10,OPLST(XR)        STORE DCB ADDRESS IN OPEN LIST(J)    00000410
          STC   FLAGR,OPLST(XR)     SET I/O FLAG IN OPEN LIST(J)         00000420
       AR       9,ARG                                                    00000430
          MVC   DCBDDNAM+2(2),0(9)                                       00000440
         MVC    DCBMACR(2),FLAG     SET MACRF IN DCB                     00000450
         LA     10,96(0,10)         BUMP REGISTER 10 TO NEXT DCB         00000460
          LA    XR,4(0,XR)          BUMP  XR TO NEXT IN OPEN LIST        00000470
         CLI    0(TMP),X'80'        LAST ARGUMENT?                       00000490
         BNE    LOOP                NO                                   00000500
         S      XR,FOUR             YES                                  00000510
          AR    XR,11                                                    00000520
          OI    0(XR),X'80'         MASK IN BIT TO STOP OPEN             00000530
         SVC    19                  TO OPEN ROUTINE (R1 POINTS TO OPLST) 00000540
         L      13,SAVE+4                                                00000550
       RETURN   (14,12)                                                  00000560
GET       SAVE  (14,12),,*                                               00000570
         BALR   12,0                                                     00000580
         USING  *,12                                                     00000590
          ST    13,SAVE+4                                                00000600
          LR     HOLD,13                                                 00000610
          LA    13,SAVE                                                  00000620
         ST     13,8(0,HOLD)                                             00000630
          L     COMN,ADCON          GET ADDRESS OF FSTAT                 00000640
          LR    XR,1                POINT XR TO ARGUMENT LIST            00000650
          L     ARG,0(0,XR)                                              00000660
         L      ARG,0(0,ARG)        GET LOGICAL UNIT NUMBER              00000670
         LPR    ARG,ARG                                                  00000680
         S      ARG,TEN                                                  00000690
       SLA      ARG,2(0)                                                 00000700
         LA     2,CHECK
         AR     2,ARG                                                    00000704
         TM     0(2),X'7F'          HAS OPEN BEEN ISSUED ?               00000706
         BNO    GETBAD                   IF NOT OPEN ABEND               00000708
          L     1,LST(ARG)          R1 POINTS TO PROPER DCB              00000710
          LR    10,1                SET 10 TO DCB                        00000720
         LA     FLAGR,1(0,0)
         XR     0,0
         L      FMT,DCBRECFM
         LTR    FMT,FMT
         BNP    STRBMP
         LA     FLAGR,2(0,0)
         LA     0,4(0,0)
STRBMP   ST     0,BLMP
```

```
          L     C,4(0,XR)           GET RECORD ADDRESS                    C0000730
          N     C,MASK
          A     0,BUMP
          ST    FLAGR,92(0,COMN)
          ST    0,36(0,COMN)                                              C0000740
          S     0,BUMP
           L    15,48(0,1)          LOAD TO GET ROUTINE                   C0000750
           BALR 14,15               LING TO GET                           C0000760
          L      TMP,96(0,COMN)     COUNT NUMBER OF RECORDS               C0000770
          LA    TMP,1(0,TMP)                                              C0000780
          ST    TMP,96(0,COMN)                                            C0000790
          L     13,SAVE+4                                                 C0000800
         RETURN (14,12)                                                   C0000810
*  GET REQUEST FOR UNOPENED OR ECF'ED LOGICAL UNIT                        C0000811
GETBAD    LA    2,TABL                                                    C0000812
          AR    2,ARG                                                     C0000813
          MVC   GETMESS+49(2),0(2)                                        C0000814
          LA    1,GETMESS                                                 C0000815
          WTO   MF=(E,(1))                                                C0000816
          ABEND 2989,DUMP,STEP                                            C0000817
EOFRET    LA    2,CHECK
          AR    2,ARG                                                     C0000824
          NI    0(2),X'00'          TURN OFF OPEN INDICATER               C0000826
          L     13,SAVE+4                                                 C0000828
          LM    2,12,28(13)                                               C0000830
          L     14,12(13)                                                 C0000840
          MVI   12(13),X'FF'                                              C0000850
           LA   15,4(0,0)                                                 C0000860
          BCR   15,14                                                     C0000870
PUT       SAVE  (14,12),,*                                                C0000880
          BALR  12,0                                                      C0000890
          USING *,12                                                      C0000900
          ST    13,SAVE+4                                                 C0000910
           LR   HOLD,13                                                   C0000920
           LA    13,SAVE                                                  C0000930
          ST    13,8(0,HOLD)                                              C0000940
          LR    XR,1                POINT XR TO ARGUMENT LIST             C0000950
          L     ARG,0(0,XR)                                               C0000960
           L    ARG,0(0,ARG)                                              C0000970
          LPR   ARG,ARG                                                   C0000980
          S     ARG,TEN                                                   C0000990
        SLA     ARG,2(0)                                                  C0001000
          LA    2,CHECK
          AR    2,ARG                                                     C0001004
          TM    0(2),X'7F'          HAS OPEN BEEN ISSUED ?                C0001006
          BNC   PUTBAD                    IF NOT ABEND                    C0001009
          L     1,LST(ARG)                                                C0001010
           L    0,4(0,XR)           LOAD R0 WITH ADDRESS OF RECORD        C0001020
WRITE     L     15,48(0,1)                                                C0001030
          BALR  14,15                                                     C0001040
```

```
          L     13,SAVE+4                                          00001050
        RETURN (14,12)                                           00001060
*  PUT REQUEST FOR UNOPENED LOGICAL UNIT                         00001062
PUTBAD   LA     2,TABL                                           00001063
         AR     2,ARG                                            00001064
         MVC    PUTMESS+39(2),0(2)                               00001065
         LA     1,PUTMESS                                        00001066
         WTO    MF=(E,(1))                                       00001067
         ABEND  2989,DUMP,STEP                                   00001068
CLOSE    SAVE   (14,12),,*                                       0000107C
         BALR   12,0                                             00001080
         USING  *,12                                             00001090
         ST     13,SAVE+4                                        0000110C
          LR    HOLD,13                                          00001110
          LA     13,SAVE                                         00001120
         ST     13,8(0,HOLD)                                     00001130
         LR     XR,1              POINT XR TO ARGUMENT LIST      00001140
         L      ARG,0(0,XR)                                      00001150
          L     ARG,0(0,ARG)                                     0000116C
         LPR    ARG,ARG                                          0000117C
         S      ARG,TEN                                          00001180
       SLA      ARG,2(0)                                         00001190
         LA     2,CHECK                                          00001194
         AR     2,ARG                                            00001196
         NI     0(2),X'00'        TURN OFF OPEN INDICATER        0000120C
         L      1,LST(ARG)                                       00001210
          ST    1,CONST                                          00001220
         OI     CONST,X'80'                                      00001230
         LA     1,CONST                                          00001240
          SVC   20                TO CLOSE                       00001250
          L     13,SAVE+4                                        00001260
        RETURN (14,12)                                           00001270
REOPEN   SAVE   (14,12),,*                                       0000128C
         BALR   12,0                                             0000129C
         USING  *,12                                             00001300
         ST     13,SAVE+4                                        0000131C
          LR    HOLD,13                                          00001320
          LA     13,SAVE                                         00001330
         ST     13,8(0,HOLD)                                     00001340
         LR     XR,1              POINT XR TO ARGUMENT LIST      00001350
         L      ARG,0(0,XR)                                      0000136C
          L     ARG,0(0,ARG)                                     00001370
         LR     TMP,ARG                                          0000138C
         LPR    ARG,ARG                                          00001390
         S      ARG,TEN                                          00001400
       SLA      ARG,2(0)                                         0000140C
         LA     2,CHECK                                          00001406
         AR     2,ARG                                            0000141C
         OI     0(2),X'7F'        TURN ON OPEN INDICATER         00001420
          L     1,LST(ARG)                                       00001430
```

```
         LR     10,1                                                    00001420
         ST     1,CONST2                                                00001430
         LTR    TMP,TMP                                                 00001440
         BP     PREO                                                    00001450
          OI    CONST2,X'8F'                                            00001460
         MVC    FLAG(2),OUT                                             00001470
          B     REO                                                     00001480
PREO     OI     CONST2,X'80'                                            00001490
         MVC    FLAG(2),IN                                              00001500
REO       LA    1,CONST2                                                00001510
         MVC    DCBMACR(2),FLAG                                         00001520
         SVC    19                    TO OPEN                           00001530
          L     13,SAVE+4                                               00001540
        RETURN (14,12)                                                  00001550
HOLD     EQU    3                                                       00001560
XR       EQU    4                                                       00001570
TMP       EQU   5                                                       00001580
ARG      EQU    6                                                       00001590
FLAGR    EQU    7                                                       00001600
COMN     EQU    8                                                       00001610
FMT      EQU    9
ADCON     DC    V(FSTAT)                                                00001620
ADEOF     DC    A(EOFRET)                                               00001630
CONST2   DC     F'0'                                                    00001640
CONST    DS     F                                                       00001650
BUMP     DS     F
MASK     DC     X'00FFFFFF'
SAVE     DS     18F                                                     00001660
LST      DC     10F'0'                                                  00001670
TABL      DS    0F                                                      00001680
          DC    X'F1F00000'                                             00001690
          DC    X'F1F10000'                                             00001700
          DC    X'F1F20000'                                             00001710
          DC    X'F1F30000'                                             00001720
          DC    X'F1F40000'                                             00001730
          DC    X'F1F50000'                                             00001740
          DC    X'F1F60000'                                             00001750
          DC    X'F1F70000'                                             00001760
          DC    X'F1F80000'                                             00001770
          DC    X'F1F90000'                                             00001780
ONE      DC      F'1'                                                   00001790
TWO      DC     F'2'                                                    00001800
FOUR     DC     F'4'                                                    00001810
TEN       DC    F'10'                                                   00001820
FLAG      DS    F                                                       00001830
LNGT     DS     F                                                       00001840
MODL     DCB    DDNAME=FT10F001,DSORG=PS,MACRF=(GM),EODAD=EOFRET        00001850
IN       DC     X'5000'                                                 00001860
OUT       DC    X'0050'                                                 00001870
CHECK    DC     10F'0'
```

```
PUTMESS  WTO   'CALL PUT FOR UNOPENED LOGICAL UNIT XX.',MF=L                00001842
         DS    F                                                          00001844
GETMESS  WTO   'CALL GET FOR UNOPENED OR EOF-ED LOGICAL UNIT XX.',MF=L    00001846
DUM       DSECT                                                          00001880
OPLST    DC    10F'-1'                                                    00001890
DCB1     DS    24F                                                        00001900
         DCBD  DSORG=(PS,DA)                                              00001910
          END                                                             00001920
SETCLK   CSECT
         SAVE  (14,12),,*
         BALR  12,0
         USING *,12
         ST    13,SAVE+4
         LR    10,13
         LA    13,SAVE
         ST    13,8(10)
        STIMER TASK,TUINTVL=MAX
          L    13,SAVE+4
       RETURN (14,12)
SAVE     DS    18F
MAX      DC    X'C4B20100'
         END
COLPRG   CSECT                                                           00000010
         ENTRY COL,COLS,ICOL,ICOLS                                       00000020
         USING *,15                                                      00000030
ICOLS    SAVE  (14,12),,*                                                00000040
HERE     OI    PLSW+1,X'F0'        SET PLURAL SWITCH YES                 00000050
         NI    INTSW+1,X'00'       SET INTEGER SWITCH YES                00000060
         NI    INSW+1,X'00'        SET INTEGER SWITCH YES                00000070
         B     RTN                                                       00000080
         USING *,15                                                      00000090
COLS     SAVE  (14,12),,*                                                00000100
         OI    PLSW+1,X'F0'        SET PLURAL SWITCH YES                 00000110
         OI    INSW+1,X'F0'        SET INTEGER SWITCH NO                 00000120
         OI    INTSW+1,X'F0'       SET INTEGER SWITCH NO                 00000130
         B     RTN                                                       00000140
         USING *,15                                                      00000150
ICOL     SAVE  (14,12),,*                                                00000160
         NI    PLSW+1,X'00'        SET PLURAL SWITCH NO                  00000170
         NI    INSW+1,X'00'        SET INTEGER SWITCH YES                00000180
         NI    INTSW+1,X'00'       SET INTEGER SWITCH YES                00000190
         B     RTN                                                       00000200
         USING *,15                                                      00000210
COL      SAVE  (14,12),,*                                                00000220
         NI    PLSW+1,X'00'        SET PLURAL SW NO                      00000230
         OI    INSW+1,X'F0'        SET INTEGER SWITCH NO                 00000240
         OI    INTSW+1,X'F0'       SET INTEGER SWITCH NO                 00000250
         B     RTN                                                       00000260
RTN      BALR  12,0                                                      00000270
         USING *,12                                                      00000280
```

```
         ST    13,HOLD                                                  00000290
         SR    CHAR,CHAR                                                00000300
         NI    ALLOWED+1,X'00'                                          00000310
         NI    HADSIGN+1,X'00'                                          00000320
         NI    BLNKSW+1,X'00'                                           00000330
         OI    DPLAST+1,X'F0'      SET DEC LAST+1 TO NOGOOD             00000340
         NI    BLAST+1,X'CC'       SET BLANK LAST+1 TO OK               00000350
         NI    CHSW+1,X'00'                                             00000360
         NI    TBSW+1,X'CC'                                             00000370
         OI    TB2SW+1,X'F0'                                            00000380
         NI    BAST+1,X'CC'                                             00000390
         OI    DNLY+1,X'FC'                                             00000400
         L     COMN,ADCON          GET FSTAT                            00000410
         L     REC,36(0,COMN)      GET CARD ADDR                        00000420
         L     LEN,0(0,1)          GET FIRST ARG ADDRESS                00000430
         L     LEN,0(0,LEN)                                             00000440
         LTR   LEN,LEN             TEST NEG ARGUMENT                    00000450
         BNP   NEGAR                                                    00000460
         S     LEN,ONE                AND ADJUST                        00000470
         AR    REC,LEN                ADDRESS                           00000480
         OI    CTSW+1,X'F0'                                             00000490
         SR    SCALE,SCALE         SET SCALE FACTOR TO 0                00000500
PLSW     BC    15,PLURAL           SWITCH FOR SINGULAR/PLURAL           00000510
SING     LA    LEN,1(0,0)          SET LENGTH TO 1                      00000520
         CLI   0(1),X'80'          TEST FOR EXTRA ARGUMENT              00000530
         BE    ONWARD              NOT FOUND                            00000540
         L     SCALE,4(0,1)        GET SCALE FACTOR                     00000550
         L     SCALE,0(0,SCALE)                                         00000560
         B     ONWARD                                                   00000570
PLURAL   LCR   LEN,LEN                                                  00000580
         L     LOC,4(0,1)                                               00000590
         A     LEN,0(0,LOC)                                             00000600
         BNP   NEGAR             NG IF LENGTH IS NEGATIVE OR ZERO       00000610
         CLI   4(1),X'80'         TEST FOR EXTRA ARGUMENT               00000620
         BE    ONWARD             NONE FOUND                            00000630
         L     SCALE,8(0,1)       GET THIRD ARGUMENT                    00000640
         L     SCALE,0(0,SCALE)                                         00000650
ONWARD   L     MASK,=X'0000000F'  TO STRIP ZONE                         00000660
         SR    COUNT,COUNT        ZERO COUNT REGISTER                   00000670
INSW     BC    15,FLINT           SWITCH FOR FLOATING                   00000680
         SR    TEMP,TEMP          INITIALIZE FOR INTEGER                00000690
         SR    TEMP2,TEMP2        CLEAR TEMP STORE                      00000700
         NI    SIGNSW+1,X'00'     SET SIGN SWITCH TO POSITIVE           00000710
         OI    DECSW+1,X'F0'      SET DECIMAP TO NOGOOD                 00000720
         NI    STORESW+1,X'00'                                          00000730
         B     GET                                                      00000740
FLINT    SDR   0,0                INITIALIZE FOR FLOATING POINT         00000750
         NI    FSIGN+1,X'00'      SET FSIGN TO POSITIVE                 00000760
         NI    DECSW+1,X'00'      SET DECIMAL TO OK                     00000770
         OI    STORESW+1,X'F0'                                          00000780
```

```
         B       GET                                                    00000790
GET      IC      CHAR,0(COUNT,REC)  GET A CHARACTER                    00000800
         LA      COUNT,1(0,COUNT)   ADD ONE TO COUNT                    00000810
         C       CHAR,ZERO                                              00000820
         BL      CHECK              TO NON-NUMERIC CHECKS               00000830
         C       CHAR,NINE                                              00000840
         BH      NOGOOD                                                 00000850
CHSW     BC      15,NOGOOD                                              00000860
         NI      ONLY+1,X'00'                                           00000870
GOOD     OI      ALLOWED+1,X'F0'    SET SIGN SWITCH TO NOGOOD           00000880
         NI      DPLAST+1,X'00'     SET DEC LAST TO OK                  00000890
TBSW     BC      15,MSK                                                 00000900
         OI      BLNKSW+1,X'F0'                                         00000910
MSK      NR      CHAR,MASK          STRIP ZONE                          00000920
INTSW    BC      15,FLOAT           FLOATING POINT SWITCH               00000930
         M       TEMP,TEN                                               00000940
         AR      TEMP2,CHAR                                             00000950
         BCT     LEN,GET            MORE?                               00000960
ISCALE   LTR     SCALE,SCALE        TEST FOR SCALE                      00000970
         BP      DIV                                                    00000980
         BZ      SIGNSW                                                 00000990
         LCR     SCALE,SCALE        MAKE POSITIVE                       00001000
MPY      M       TEMP,TEN                                               00001010
         BCT     SCALE,MPY                                              00001020
SIGNSW   BC      15,NEGAT                                               00001030
         LR      0,TEMP2            STORE ANS                           00001040
OUT      L       13,HOLD                                                00001050
         RETURN  (2,12)                                                 00001060
NEGAT    LCR     0,TEMP2            MAKE SIGN MINUS                     00001070
         B       OUT                                                    00001080
DIV      D       TEMP,TEN                                               00001090
         SR      TEMP,TEMP          CLEAR REMAINDER                     00001100
         BCT     SCALE,DIV                                              00001110
         B       SIGNSW                                                 00001120
FLOAT    STC     CHAR,EXP+3                                             00001130
         MD      0,FTEN                                                 00001140
         AD      0,EXP                                                  00001150
CTSW     BC      15,TEST                                                00001160
         LA      SCALE,1(0,SCALE)                                       00001170
TEST     BCT     LEN,GET                                                00001180
FSCALE   LTR     SCALE,SCALE        TEST FOR SCALE FACTOR               00001190
         BZ      FSIGN                                                  00001200
         BM      FMULO                                                  00001210
FDIV     DD      0,FTEN                                                 00001220
         BCT     SCALE,FDIV                                             00001230
FSIGN    BC      15,FNEG                                                00001240
         B       OUT                                                    00001250
FNEG     LCDR    0,0                MAKE MINUS                          00001260
         B       OUT                                                    00001270
FMULO    LCR     SCALE,SCALE        MAKE POSITIVE                       00001280
```

```
FMUL     MD    0,FTEN                                                00001290
         BCT   SCALE,FMUL                                            00001300
         B     FSIGN                                                 00001310
CHECK    C     CHAR,BLANK                                            00001320
         BNE   CHECK2                                                00001330
BLNKSW   BC    15,NOGOOD          BLANK NOT ALLOWED                  00001340
TB2SW    BC    15,MORE                                               00001350
         OI    CHSW+1,X'FC'       SET CHARSW TO NG                   00001360
MORE     BCT   LEN,GET            MORE?                              00001370
BLAST    BC    15,NOGOOD                                             00001380
BAST     BC    15,DNLY                                               00001390
         L     0,20(0,COMN)       FIELD WAS ALL BLANK                00001400
         A     0,ONE                                                 00001410
         ST    0,20(0,COMN)       NBLANK                             00001420
         L     0,28(0,COMN)       BORBLK                             00001430
         LE    0,28(0,COMN)       FLOAR RETURN                       00001440
         B     OUT                                                   00001450
DNLY     BC    15,NOGOOD                                             00001460
         B     FSCALE                                                00001470
CHECK2   C     CHAR,DECPT         DECIMAL?                           00001480
         BNE   CHECK3                                                00001490
DECSW    BC    15,NOGOOD          DECIMAN NOT ALLOWED                00001500
         OI    DECSW+1,X'F0'      SET DECSW TO NOGOOD                00001510
         OI    ALLOWED+1,X'FC'    SET NO SIGN AFTER DEC POINT        00001520
         NI    BLNKSW+1,X'00'     SET BLANK SWITCH TO GOOD           00001530
         OI    TBSW+1,X'FC'                                          00001540
         NI    TB2SW+1,X'C0'                                         00001550
         NI    BLAST+1,X'00'      SET BLANK LAST TO OK               00001560
         OI    BAST+1,X'FC'                                          00001570
         SR    SCALE,SCALE        SET SCALE FACTOR TO ZERO           00001580
         NI    CTSW+1,X'00'       SET PLACE SWITCH TO COUNT NEW SCALE 00001590
         BCT   LEN,GET                                               00001600
DPLAST   BC    15,NOGOOD                                             00001610
         B     FSIGN                                                 00001620
CHECK3   C     CHAR,MINUS         MINUS SIGN?                        00001630
         BNE   CHECK4                                                00001640
         OI    SIGNSW+1,X'F0'     YES, SET SIGN SWITCHES             00001650
         OI    FSIGN+1,X'F0'                                         00001660
         B     ALLOWED                                               00001670
CHECK4   C     CHAR,PLUS          PLUS SIGN?                         00001680
         BNE   CHECK5             NO                                 00001690
ALLOWED  BC    15,NOGOOD          IS LIGN LEGIT?                     00001700
         OI    ALLOWED+1,X'F0'    SET SWITCH TO NG AFTER FIRST SIGN  00001710
         OI    BLAST+1,X'F0'      SET BLANK LAST TO NOGOOD           00001720
         OI    HADSIGN+1,X'F0'    SET SWITCH IN LAST CHAR CHECK      00001730
         BCT   LEN,GET                                               00001740
         BC    15,NOGOOD          IF SIGN IN LAST POS NOGOOD         00001750
CHECK5   C     LEN,ONE            LAST CHARACTER?                    00001760
         BNE   NOGOOD                                                00001770
         CLC   12(4,COMN),ONE     YES. MD650 ON?                     00001780
```

```
         BNE   NOGOOD                                                  00001790
HADSIGN  BC    15,NCGOCD           PREVIOUS SIGN                      00001800
         ST    CHAR,BIN                                               0000181C
         N     CHAR,=X'000000F0'   STRIP NUMERIC                      00001820
         C     CHAR,=X'CCCCCCDC'   MINUS?                             00001830
         BNE   ZONEP                                                  00001840
         OI    SIGNSw+1,X'FO'      YES                                0000185C
         OI    FSIGN+1,X'FC'                                          00001860
         B     CHECK6                                                 0000187C
ZONEP    C     CHAR,=X'0CC000C0'   PLUS ZONE?                         00001880
         BNE   NOGCCD                                                 00001890
CHECK6   L     CHAR,BIN                                               00001900
         NR    CHAR,MASK                                              00001910
         C     CHAR,=X'CCCCCCC9'                                      00001920
         BNH   GOOD                                                   00001930
NOGOOD   L     0,16(C,CCMN)                                           00001940
         A     0,ONE                                                  00001950
         ST    C,16(0,CCMN)        NCLEER                             00001960
         STE   0,84(0,COMN)                                           00001970
         MVC   112(4,COMN),96(COMN)                                   00001980
         L     0,32(0,CCMN)        FORERR                             00001990
         LE    0,32(0,COMN)        FORERR                             00002000
NG2      STC   CHAR,24(0,CCMN)     ACTEER                             00002010
         S     COUNT,ONE                                              00002020
         ST    COUNT,80(0,COMN)    NCGOCH                             00002030
STORESw  BC    15,OUT                                                 00002040
         ST    TEMP2,84(0,CCMN)                                       00002050
         B     OUT                                                    00002060
NEGAR    L     0,128(C,CCMN)       GET NEGARG                         00002070
         A     0,ONE                                                  00002080
         ST    C,128(0,CCMN)                                          00002090
         L     C,32(0,CCMN)        FCRERR                             00002100
         LE    0,32(0,CCMN)                                           00002110
         B     OUT                                                    00002120
TEMP     EQU   2                                                      00002130
TEMP2    EQU   3                                                      00002140
COMN     EQU   4                                                      00002150
REC      EQU   5                   ADDRESS OF FIRST CHAR              0000216C
LEN      EQU   6                   LENGTH OF FIELD                    0000217C
SCALE    EQU   7                                                      0000218C
CHAR     EQU   8                                                      00002190
SIGN     EQU   9                                                      00002200
MASK     EQU   10                                                     00002210
COUNT    EQU   11                                                     00002220
LOC      EQU   13                                                     00002230
FTEN     DC    D'10.0'                                                00002240
EXP      DC    X'46CCCCCCCCCCCCCC'                                    00002250
ADCON    DC    V(FSTAT)                                               00002260
         DS    0F                                                     00002270
ZERO     DC    X'CCCCCCF0'                                            00002280
```

```
NINE     DC    X'CCCCCCF9'                                  00002290
BLANK    DC    X'CCCCCC40'                                  00002300
PLUS     DC    X'CCCCCC4E'                                  00002310
MINUS    DC    X'CCCCCC6C'                                  00002320
DECPT    DC    X'CCCCCC4B'                                  00002330
TEN      DC    F'10'                                        00002340
ONE      DC    F'1'                                         00002350
HOLD     DS    F                                            00002360
BIN      DS    F                                            00002370
         END                                                00002380
NUMARG   CSECT
         SAVE  (14,12),,*
         BALR  12,0
         USING *,12
         L     3,4(13)
         L     4,24(3)
         LA    5,4(0,0)
         SR    6,6
         LA    7,1(0,0)
LOOP     AR    6,7
         CLC   0(1,4),=X'80'
         BE    OUT
         AR    4,5
         B     LOOP
OUT      LR    0,6
         L     5,0(0,1)
         ST    6,0(0,5)
       RETURN (2,12)
         END
ITIME    CSECT
         SAVE  (14,12),,*
         BALR  12,0
         USING *,12
         ST    13,SAVE+4
         LR    10,13
         LA    13,SAVE
         ST    13,8(13)
      TTIMER
         L     13,SAVE+4
      RETURN (1,12)
SAVE     DS    18F
         END
DATE     CSECT
         SAVE  (14,12),,*
         BALR  12,0
         USING *,12
         L     ARG,0(0,1)
         ST    13,SAVE+4
         LR    10,13
         LA    13,SAVE
```

```
         ST    13,8(10)
         TIME  DEC
         ST    0,TIME
         ST    1,JULD
         MVC   ANS(20),PAT
         MVC   DPM+2(2),D28
         MVI   TIME+3,X'C0'
         ED    ANS+1(9),TIME
         SP    MONTH,MONTH
         ZAP   WORK,JULD
         DP    WORK,TEN3
         ZAP   DAY,WORK+5(3)
         ZAP   DATER,YEAR+3(2)
         DP    YEAR,FOUR
         CP    YEAR+4(1),ZERO
         BNE   NOT
         MVC   DPM+2(2),D29
NOT      LA    XREG,DPM
         LA    MTHS,12(0,0)
LOOP     AP    MONTH,ONE
         SP    DAY,0(2,XREG)
         BNP   GOTIT
         LA    XREG,2(0,XREG)
         BCT   MTHS,LOOP
ERROR    MVI   ANS,X'FC'
         MVC   ANS+1(19),ANS
         B     OUT
GOTIT    AP    DAY,0(2,XREG)
         MP    DAY,TEN2
         AP    DATER,DAY
         MP    MONTH,TEN4
         AP    DATER,MONTH
         ED    ANS+10(10),DATER
OUT      MVC   0(20,ARG),ANS
         L     13,SAVE+4
        RETURN (14,12)
XREG     EQU   3
MTHS     EQU   4
ARG      EQU   5
TIME     DS    F
JULD     DS    F
         DC    X'4040404040'
ANS      DS    5F
PAT      DC    X'4040202070A2020 7A2020402020206120206120 20'
DATER    DC    PL4'0'
MONTH    DC    PL5'0'
WORK     DS    0D
YEAR     DC    PL5'0'
DAY      DC    PL4'0'
FOUR     DC    P'4'
```

```
ZERO     DC    P'0'
TEN2     DC    P'100'
TEN3     DC    P'1000'
TEN4     DC    P'10000'
D29      DC    P'29'
DPM      DC    P'31',P'28',P'31',P'30',P'31',P'30',P'31',P'31'
         DC    P'30',P'31',P'30',P'31'
ONE      DC    P'1'
D28      DC    P'28'
SAVE     DS    18F
         END
```